# An adaptive strategy based on linear prediction of queue length to minimize congestion in Barabási–Albert scale-free networks*

Shen Yi(沈 毅)†

*College of Information Science and Technology, Nanjing Agricultural University, Nanjing* 210095, *China*

(Received 27 August 2012; revised manuscript received 16 October 2012)

In this paper, we propose an adaptive strategy based on the linear prediction of queue length to minimize congestion in Barabási–Albert (BA) scale-free networks. This strategy uses local knowledge of traffic conditions and allows nodes to be able to self-coordinate their accepting probability to the incoming packets. We show that the strategy can delay remarkably the onset of congestion and systems avoiding the congestion can benefit from hierarchical organization of accepting rates of nodes. Furthermore, with the increase of prediction orders, we achieve larger values for the critical load together with a smooth transition from free-flow to congestion.

**Keywords:** linear prediction, congestion, networks

**PACS:** 89.75.Hc, 02.50.–r **DOI:** 10.1088/1674-1056/22/5/058902

## 1. Introduction

Traffic flows in real complex systems such as the Internet, the power grid, and transportation networks have recently attracted a lot of attention.[1–6] It is shown that the optimization of information or material flow on these systems is a very important issue for ensuring their stability. Moreover, the structure of these complex systems can be efficiently described by a graph[4–8] and typically shows a scale-free distribution.[4] Therefore, it allows us to use modern complex network theory[8,9] together with tools inherited from statistical physics[10–12] to study the dynamical properties of such complex systems, especially the realization of free traffic flow on them.

In recent years, many investigations have tackled the critical properties of the congestion transitions[13–21] of networks. These studies have focused on designing efficient routing strategies for providing short delivery times and avoiding congestion. It has been shown that the network topologies and the load of information on it have a great influence on the design of efficient routing strategies.[13–16] Regarding the influence of network topologies, a number of routing mechanisms have been studied on different structures,[14–16] and indicate that homogeneous networks can bear more traffic because of the absence of high betweenness nodes.[19] However, changing the network topology is always expensive. Therefore, it is reasonable to develop better routing strategies to enhance the network capacity. Examples of such strategies are biased random walks,[18] the shortest path,[19,20] the efficient path,[21] and so on. These routing strategies can be conveniently reformulated to incorporate information about the congestion state of nodes. In these strategies, congestion-aware schemes have been used and allow dynamically changing the paths in order to bypass those over-congested nodes. In addition to the design of efficient routing strategies, several strategies such as the implementation of cutting links between large-degree nodes[22] and packet-dropping mechanisms[23] for avoiding the congestion of single nodes have been also proposed recently. However, the strategies proposed above usually originate from a static perspective that the mechanisms to avoid congestion are static (i.e., packets are suggested to be forwarded using fixed routing strategies). As a matter of fact, the optimal routing policy depends strongly on the state of congestion of the system.[24,25] Furthermore, the traffic between nodes is dynamic and the topology of networks may evolve with time. Therefore, in order to enhance the network capacity, it seems appropriate to allow the nodes to use local knowledge of traffic conditions and self-coordinates to avoid congestion under the instant traffic conditions.

In this paper, we propose an adaptive strategy based on linear prediction of queue length to minimize congestion in Barabási–Albert (BA) scale-free networks. In this strategy, we allow nodes to be able to predict their queue length and use an adaptive evolution function to decide whether or not to accept packets that flow from their neighbors. The local knowledge of traffic conditions and the ability for nodes to self-coordinate their accepting probability to the incoming packets are employed in this strategy. We show that the strategy can delay remarkably the onset of congestion and systems avoiding the congestion can benefit from hierarchical organization of accepting rates of nodes. Furthermore, with the increase of pre-

*Project supported by the National Natural Science Foundation of China (Grant No. 60672095), the Fundamental Research Funds for the Central Universities of China (Grant No. KYZ201300), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK2013000), and the Youth Sci-Tech Innovation Fund of Nanjing Agricultural University, China (Grant No. KJ2010024).

†Corresponding author. E-mail: shen_yi1979@njau.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd 

diction orders we achieve larger values for the critical load together with a smooth, second-order-like, transition from free-flow to congestion.

## 2. Traffic model of the adaptive strategy

In this model, we consider the transfer of information packets between adjacent nodes as a probabilistic event.[26] The probability for the packet to go from node $i$ to its neighbor $j$ is written as $p_{ij} = 1/k_i$, where $k_i$ is the degree of node $i$. When the packet arrives at its destination, the packet is removed from the system. Here, we use a set of dynamic equations for describing the queue length of nodes $q_i^t$ $(i = 1,2,\ldots,N)$ at time $t$, where $N$ is the number of nodes in the network. At each time step and with probability $p$, a new packet is generated on node $i$ with a randomly chosen destination. In each time step from $t$ to $t+1$, node $i$ tries to send a packet in its queue to any of its neighbors and the packet will be accepted with probability $\lambda$. If the packet is accepted by a neighbor, it will be removed from the queue of node $i$. We assume that all the nodes can store as many packets as needed but can deliver a finite number of them at each time step. We set this number to be 1 for simplicity.

At time $t$, the incoming flow of packets that arrive at the queue of node $i$ is

$$q_{\text{in}}^t = \lambda_i^t \sum_{j=1}^{N} \theta(q_j^t) A_{ji} p_{ji}, \tag{1}$$

and the probability that a packet from node $i$ is delivered to the neighbors of node $i$ is

$$q_{\text{out}}^t = \sum_{j=1}^{N} \lambda_j^t \theta(q_i^t) A_{ij} p_{ij}, \tag{2}$$

where $\theta(x)$ is the Heaviside step function: $\theta(x) = 1$ if $x > 0$ and $\theta(x) = 0$ otherwise; $A_{ij}$ is the element in the adjacency matrix $\boldsymbol{A}$ which is defined as $A_{ij} = 1$ if nodes $i$ and $j$ are connected, and $A_{ij} = 0$ otherwise. Following the above ingredients, we can easily write the time-discrete Markov chain of the traffic model as

$$q_i^{t+1} = q_i^t + p + q_{\text{in}}^t - q_{\text{out}}^t. \tag{3}$$

The congestion will appear when the system is unable to balance the incoming flow of new packets with a successful delivery of old ones.[17,27] Therefore, in order to avoid the congested phase[17] of a network, the $\lambda_i^t$ should aim at operating in the adverse case (which is called free-flow regime[17] of a network). To this aim, each node should choose its own accepting probability $\lambda_i^t$ and attempt to reach an optimal queue length. For each node, it can store all the packets flowing from its neighbors, and at the same time, with probability $p$, a new packet is generated on it. Therefore, the optimal queue length is set to be $(1+\langle k\rangle)\,p$, where $\langle k\rangle$ is the average degree, so that the traffic is homogeneously distributed across the network.

Based on linear prediction of queue length, we propose the adaptive evolution equations $\lambda_i^t$ for node $i$ as

$$\lambda_i^{t+1} = \exp\left(-\beta \hat{q}_i^{t+1}\right), \tag{4}$$

where $\beta$ is the parameter that controls the velocity of adjusting accepting probability, and $\hat{q}_i^{t+1}$ is the $L$-order predictive value of $q_i^{t+1}$.

According to Eq. (4), the under-congested node ($\hat{q}_i^{t+1} \to 0$) will tend to total acceptance ($\lambda_i^{t+1} \to 1$), whereas those congested nodes ($\hat{q}_i^{t+1} \to +\infty$) will tend to total rejection ($\lambda_i^{t+1} \to 0$). Here, $\lambda_i^{t+1}$ is set to be 0.5 whenever $\hat{q}_i^{t+1} = (1+\langle k\rangle)\,p$ for generality, then $\beta = \ln 2/\left((1+\langle k\rangle)\,p\right)$. Based on the theory of linear prediction,[28] $\hat{q}_i^{t+1}$ can be written as a linear combination of the previous queue length of node $i$,

$$\hat{q}_i^{t+1} = -\sum_{k=1}^{L} a_k q_i^{t+1-k}, \tag{5}$$

where $a_k$ are the predicting coefficients, and $q_i^{t+1-k}$ are the previous queue length of node $i$ at time $t+1-k$. If we obtain the actual value of $q_i^{t+1}$, the error between $q_i^{t+1}$ and its predictive value $\hat{q}_i^{t+1}$ can be calculated as

$$\begin{aligned} e(t) &= q_i^{t+1} - \hat{q}_i^{t+1} = q_i^{t+1} + \sum_{k=1}^{L} a_k q_i^{t+1-k} \\ &= \sum_{k=0}^{L} a_k q_i^{t+1-k}, \end{aligned} \tag{6}$$

where $a_0$ is set to be 1 for avoiding a trivial solution for minimizing $e(t)$. Then the Yule–Walker equations for calculating the predicting coefficients can be written as $\boldsymbol{Ra} = [1,0,\ldots,0]^{\mathrm{T}}$. The elements of $\boldsymbol{R}$ are $r_{ij} = R(i-j)$, where $R$ is the self-correlation of queue length of node $i$ and defined as $R(m) = E\left\{q_i^t \cdot q_i^{t-m}\right\}$. The $\boldsymbol{R}$ is a symmetrical matrix, so that the fast Levinson recursion method[29] can be employed to calculate $\boldsymbol{a}$.

## 3. Experiments

The structure of many real networks is far from being purely random. Quite the contrary, they typically show a scale-free (SF) distribution.[4,27,30] We start the simulations by adopting the well-known Barabási–Albert (BA) scale-free network model.[29]

This model is defined with two steps as follows.

**Growth** At the beginning, there are $m_0$ initial nodes. At each time step, a new node is added into the network, and this new node will connect to $m$ $(m \le m_0)$ different nodes which are already present in the network.

**Preferential attachment** Node $i$ will link to the new node with probability $P(k_i) = k_i / \sum_{j=1} k_j$, where $k_i$ is the degree

of node $i$. This model will generate a scale-free network with $t+m_0$ nodes and $mt$ edges in $t$ steps, and the average degree of the BA network is approximately $2m$.

### 3.1. Phase transition

All nodes are assumed to be able to generate and deliver packets. At each time step and with probability $p$, a new packet is generated on a node with a randomly chosen destination. When the packets arrive at their destination, they are removed from the network. We are interested in the critical value $p_c$ at which a phase transition takes places from free flow to congested state. This critical value can best reflect the maximal capacity of a system handing its traffic. It is known[21,31] that for $p < p_c$, the system is able to balance the numbers of created and delivered packets. In this case, the sum of all the values of queue length $Q(t) = \sum_{i=1}^{N} q_i^t$ fluctuates around a finite value. For $p > p_c$, traffic congestion occurs as $Q(t)$ increases with time. We use the following order parameter $H$[17] to monitor the phase transition:

$$H(t) = \lim_{\Delta t \to \infty} \frac{1}{Np} \frac{Q(t+\Delta t) - Q(t)}{\Delta t}, \tag{7}$$

where $H$ is in a range from 0 to 1, and $N$ is the number of nodes in the network. $H = 0$ means that the system is in the free-flow phase, while $H > 0$ means that the system is in the congested phase.

We generate a BA network with size $N = 2000$ and average degree $\langle k \rangle = 4$ for our tests. For the non-adaptive case, the accepting probability is fixed to be 1, i.e., $\lambda = 1$. In our strategy, the accepting probability $\lambda_i^t$ is set automatically according to the traffic state (the length of queue) of node $i$. Figure 1 shows the testing results. A set of values of prediction order $L$ are investigated. For clarity, we plot the results only for the values of $L = 1, 2, 4, 8, 16$.

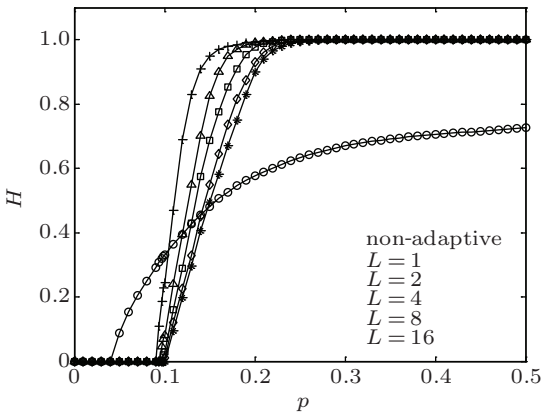



**Fig. 1.** The order parameter $H$ versus $p$ for our strategy (for different values of $L$) and random walks. All datum points are obtained by averaging over $10^6$ independent runs.

In our strategy, nodes can increase or decrease their own accepting probability according to their own traffic states (the length of their queues). This mechanism can redistribute the flow of packets and improve the capacity of the whole network. Therefore, as we can see from Fig. 1, the critical point $p_c$ of our strategy is much larger than that of the non-adaptive case. In our strategy, since a node decreases its accepting probability quickly with the increase of its queue length, the abrupt phase transition from the free flow to the congested state occurs. We increase $L$ from 0 to 25, which is large enough to investigate the phase transition of our strategy. We observe that, the critical point $p_c$ increases slowly with $L$ increasing, and the transition curve becomes smooth. The limit of $p_c$ and phase transition occur quickly.

### 3.2. Influence of prediction order on congestion

In this section, we investigate in detail the influence of $L$ on the critical point $p_c$ of our strategy. We define a coefficient denoted by $s$, which measures the range from the point of $p_c$ to the point when $H$ arrives at 0.8. In this range, the transition curves show a linear increase, and clear discrimination can be seen between the curves for different $L$, as shown in Fig. 1. The parameter $s$ is used to show how smooth the curves of $H$ is. The testing BA network has size $N = 2000$ and average degree $\langle k \rangle = 4$. Figure 2(a) shows the critical point $p_c$ as a function of $L$. Figure 2(b) shows $s$ as a function of $L$. As shown in Fig. 2, for small $L$, $p_c$ and $s$ increase quickly with $L$ increasing. However, the increment of the two parameters decreases and is almost zero for large $L$. The reason is that although larger $L$ corresponds to more accurate predictive value of the queue length, the increment of accuracy is limited for each larger $L$. The testing results show that the small value of $L$ has already allowed us to obtain obvious improvement on $p_c$ and smooth phase transition.

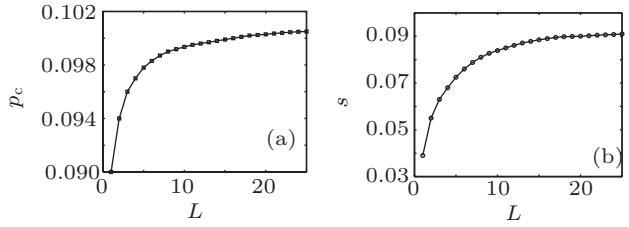



**Fig. 2.** Influence of prediction orders. (a) The critical point $p_c$ as a function of $L$; (b) the value of $s$ as a function of $L$. All datum points are obtained by averaging over $10^6$ independent runs.

### 3.3. Distribution of accepting probability

To have a deeper insight into the microscopic configurations that allow delaying the onset of congestion, we show in Fig. 3 the distribution of average accepting rates $\langle \lambda \rangle_k$ of nodes with degree $k$ as a function of degree of node. The testing BA network has size $N = 2000$ and average degree $\langle k \rangle = 4$. The prediction order $L = 8$ is adopted here since obvious improvement on $p_c$ and smooth phase transition have already been obtained, as shown in Fig. 2. From Fig. 3 we can see that the

correlation between $\langle\lambda\rangle_k$ and degrees $k$ is clear since all the nodes within the same degree-class display similar accepting rates. We observe that for $p = 0.01$ the system self-organizes homogeneously around $\langle\lambda\rangle = 0.6$. However, as $p$ increases, the accepting rate of low-degree class increases, while node with large degree starts to decrease their accepting rates progressively. The reason lies in that the hub nodes are usually more important and bear more traffic load. With the increase in $p$, low accepting probability for hub nodes can lead to the redistribution of traffic loads along edges, which can enhance the capacity of the whole network. We can see that a hierarchical (degree-based) organization of the accepting rates is strongly beneficial to avoiding the congestion of the system. Nodes with larger degrees have lower accepting rates. In other words, hub nodes in the network self-coordinate dynamically with different accepting rates to avoid congestion, which results in a remarkable decrease of network congestion.

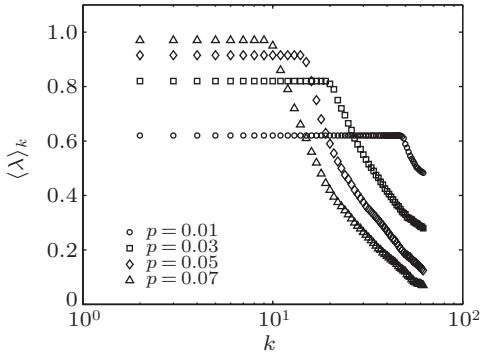



**Fig. 3.** Distributions of average accepting rate $\langle\lambda\rangle_k$ as a function of node degree. All datum points are obtained by averaging over $10^6$ independent runs.

### 3.4. Comparison between our method and other existing methods

In the above tests of our method, packets are forwarded to their destinations based on random walks. Hence, our method is a complete local method that does not need to have the knowledge of the network in advance. In this section, we compare our method with two existing global routing strategies, the efficient path strategy (EPS)[21] and the optimal path strategy (OPS),[32] in the performance of minimizing congestion. To make fair comparisons, we revise our method by using the global knowledge of shortest path, i.e., packets are forwarded in preference on their shortest path to the destination and the local adaptive strategy is used to avoid congestion at the same time. In other words, if a packet from the current node is not accepted by the following neighbor node on the shortest path, the packet may be accepted by the other neighbor of the current node. Then, the next hop of the packet is in preference on the shortest path from the current neighbor node to the original destination. The process runs until the packet arrives at the original destination. In the tests, the number of data packets that a node can forward to its neighbors in each time step is assumed to be 1 for all the methods.

In EPS, for any path from node $i$ to node $j$, as $i, x_0, x_1, \ldots, x_n, j$, we have[21]

$$L((P \to j) : \beta) = \sum_{i=0}^{n-1} k(x_i)^{\beta}, \tag{8}$$

where $\beta$ is a tunable parameter, and $k(x_i)$ is the degree of node $x_i$. The efficient routing path between node $i$ and node $j$ corresponds to the route that makes the sum $L$ a minimum. The idea of the EPS is that in order to make the load distributed evenly, packets should avoid the heavily linked nodes since they are prone to be congested. Simulation results show that the performance of minimizing congestion depends strongly on $\beta$. Therefore, before using the method, one must calculate the optimal $\beta$ at first. Furthermore, the values of $\beta$ usually are different for different topology networks.

The OPS is a heuristic method which depends on finding the exactly optimal betweenness $B^*_{\max}$ for the efficient paths. Finding $B^*_{\max}$ is mathematically equivalent to solving the problem of minimizing the sparse vertex separator which has been shown to be an NP-hard problem.[33] Thus, it is generally hard to find the exact solution of this problem within practical calculation times. It is shown that a nearly-optimal solution for $B^*_{\max}$ is found by the OPS. Although a very small and nearly-optimal limit value of $B^*_{\max}$ is obtained, it takes a great many steps for the convergence of $B_{\max}(t)$ relative to the system size by the algorithm. Therefore, this method has high computation cost.

Next, we investigate the average path length $L_{\text{ave}}$, the average traveling time $T_{\text{ave}}$, and the order parameter $H$ for all these methods. Prediction order $L = 8$ is adopted for our strategy. In Fig. 4, we show the values of average path length $L_{\text{ave}}$ as a function of the total number of nodes $N$ for three strategies. This figure shows that the values of $L_{\text{ave}}$ of our method are smaller than those of OPS, but slightly larger than those of EPS for the same sized network. Furthermore, we can observe that in the range $N \in [100, 1000]$, $L \sim \ln N$ is maintained in our method.

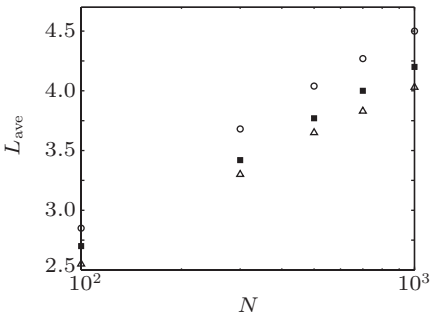



**Fig. 4.** Average path length $L_{\text{ave}}$ over 20 networks as a function of $N$ for $\langle k\rangle = 8$, for the EPS (triangles), OPS (circles), and our strategy (squares).

In Fig. 5, we show the curves of average traveling time of a packet $T_{\text{ave}}$ as a function of packet generation rate $p$. With

small $p$, we can see that the value of $T_{\rm ave}$ for our strategy is lower than that for OPS but higher than that for EPS. With the increase of $p$, we observe that for our strategy, the $p_{\rm c}$ where $T_{\rm ave}$ begins to increase intensely is larger than that for EPS but smaller than that for OPS. Although the value $T_{\rm ave}$ of our strategy is a little longer than that of EPS in the free-flow state, our strategy is proper for the tolerance of congestion because the $T_{\rm ave}$ of our strategy is much shorter than that of EPS in the congestion state.

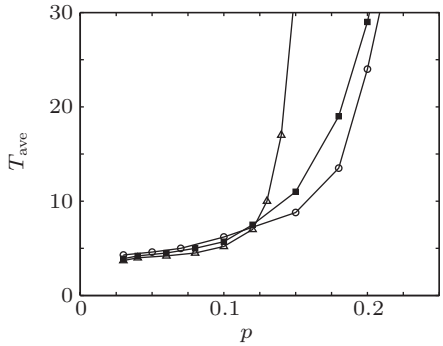



**Fig. 5.** Average traveling time over 20 networks of a packet $T_{\rm ave}$ as a function of $p$ for $N = 500$ and $\langle k\rangle = 8$ for the EPS (triangles), OPS (circles), and our strategy (squares).

In Fig. 6, we show curves of the order parameter $H$ for the EPS, OPS, and our strategy as a function of the packet generation rate $p$ for the BA networks with $\langle k\rangle = 8$ and $N = 500$. From Fig. 6, we observe that the phase transition from free-flow to congestion of our strategy is smoother than those of the other two methods, and the order parameter $H$ is larger than that of EPS. Although OPS has the largest $H$ in these methods, it takes a great many iteration steps for the convergence of $B_{\rm max}(t)$ relative to the network size, which is highly computational in practical applications.

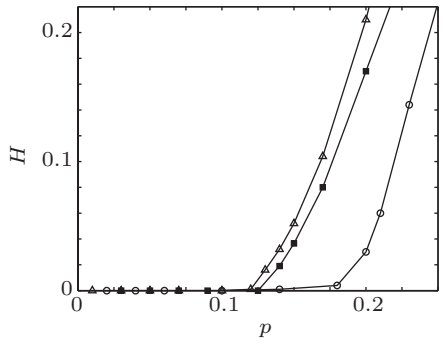



**Fig. 6.** Order parameter $H$ (averaged over 20 networks) versus $p$ for the EPS (triangles), OPS (circles), and our strategy (squares) with $\langle k\rangle = 8$ and $N = 500$.

## 4. Conclusions

In this paper, we propose an adaptive strategy to minimize congestion in BA scale-free networks. This strategy is based on linear-prediction of queue length. In the strategy, we allow nodes to be able to change their accepting probability according to their own queue length. The local knowledge of traffic conditions and the ability for nodes to self-coordinate their accepting probability to the incoming packets are used. In the experiments, we obtain a variety of dynamical behaviours. We analyse the phase transition of the strategy and show that our strategy can delay remarkably the onset of congestion. Then, the influence of prediction order on network congestion is investigated. We show that with the increase of prediction order $L$, the curve of phase transition becomes smooth and the value of critical point $p_{\rm c}$ increases. We also investigate the distribution of average accepting probability versus the degree of node, and show that a hierarchical (degree-based) organization of the accepting rates by the system is strongly beneficial for avoiding the congestion of the system. Furthermore, we compare our method with several existing methods in average path length $L_{\rm ave}$, average traveling time $T_{\rm ave}$, and the order parameter $H$. The experimental results clearly expose the advantages and disadvantages of our method.

## References


[1] Faloutsos M, Faloutsos P and Faloutsos C 1999 *Comput. Commun. Rev.* **29** 251
[2] Bono F, Gutiérrez E and Poljansek K 2010 *Physica A* **389** 5287
[3] Tadić B, Rodgers G J and Thurner S 2007 *Int. J. Bif. Chaos Appl. Sci. Eng.* **17** 2363
[4] Boccaletti S, Latorab V, Moreno Y, Chavez M and Hwang D U 2006 *Phys. Rep.* **424** 175
[5] Li S M, Xu X H and Meng L H 2012 *Chin. Phys. B* **21** 088901
[6] Wang D L, Yu Z G and Anh V 2012 *Chin. Phys. B* **21** 080504
[7] Zou A Q, Li Y and Wu J 2010 *Chin. Phys. Lett.* **27** 068901
[8] Newmam M E J 2010 *Networks: an Introduction* (Oxford: Oxford University Press)
[9] Newman M E J 2003 *SIAM Rev.* **45** 167
[10] Albert R and Barabási A L 2002 *Rev. Mod. Phys.* **74** 48
[11] Li W, Yang J Y and Hadden W C 2009 *Europhys. Lett.* **88** 68007
[12] Sun J and Avraham D B 2010 *Phys. Rev. E* **82** 0161109
[13] Sreenivasan S, Cohen R, Lopez E, Toroczkai Z and Stanley H E 2007 *Phys. Rev. E* **75** 036105
[14] Guimerà R, Arenas A, Guilera A D and Giralt F 2002 *Phys. Rev. E* **66** 026704
[15] Rosato V, Issacharoff L, Meloni S, Caligiore D and Tiriticco F 2008 *Physica A* **387** 1689
[16] Scellato S, Fortuna L, Frasca M, Gardeñes J G and Latora V 2010 *Eur. Phys. J. B* **73** 303
[17] Guimerà R, Guilera A D, Redondo F V, Cabrales A and Arenas A 2002 *Phys. Rev. Lett.* **89** 248701
[18] Wang W X, Wang B H, Yin C Y, Xie Y B and Zhou T 2006 *Phys. Rev. E* **73** 026111
[19] Goh K I, Kahng B and Kim D 2001 *Phys. Rev. Lett.* **87** 278701
[20] Echenique P, Gardeñes J G and Moreno Y 2004 *Phys. Rev. E* **70** 056105
[21] Yan G, Zhou T, Hu B, Fu Z Q and Wang B H 2006 *Phys. Rev. E* **73** 046108
[22] Liu Z, Hu M B, Jiang R, Wang W X and Wu Q S 2007 *Phys. Rev. E* **76** 037101
[23] Huang W and Chow T W S 2009 *Chaos* **19** 043124
[24] Kim K, Kahng B and Kim D 2009 *Europhys. Lett.* **86** 58002
[25] Petri G, Jensen H J and Polak J W 2009 *Europhys. Lett.* **88** 20010
[26] Noh J D 2004 *Phys. Rev. Lett.* **92** 118701
[27] Hua D Y and Wang L Y 2010 *Chin. Phys. Lett.* **27** 098901
[28] Oppenheim A V, Willsky A S and Nawab S H 1997 *Signals and Systems* (London: Pearson Higher Isia Education Press)
[29] Biggs N L 1974 *Algebraic Graph Theory* (Cambridge: Cambridge University Press)
[30] Shen Y, Pei W J, Wang K, Li T and Wang S P 2008 *Physica A* **387** 6663
[31] Albert R and Barabási A L 1999 *Science* **286** 509
[32] Danila B, Marsh J A and Bassler K E 2007 *Chaos* **17** 026102
[33] Sreenivasan S, Cohen R, López E, Toroczkai Z and Stanley H E 2007 *Phys. Rev. E* **75** 036105