Physica A 499 (2018) 472–480

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Cascade phenomenon against subsequent failures in complex networks

Zhong-Yuan Jiang [a,b,*], Zhi-Quan Liu [c,*], Xuan He [d,e], Jian-Feng Ma [a,b]

[a] School of Cyber Engineering, Xidian University, Xi'an, Shaanxi 710071, China
[b] Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an, Shaanxi 710071, China
[c] College of Cyber Security, Jinan University, Guangzhou, Guangdong 510632, China
[d] Key Laboratory of Universal Wireless Communications, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[e] China Unicom Broadband Online Limited Corporation, Beijing 100031, China

HIGHLIGHTS

- This work evaluated the network robustness against subsequent failures from a very novel perspective.
- Four heuristic key nodes discovering methods were proposed.
- Extensive simulations were done in both scale-free and random networks to verify the effectiveness of the proposed methods.

ARTICLE INFO

Article history:
Received 8 September 2017
Received in revised form 17 November 2017
Available online 16 February 2018

Keywords:
Cascade phenomenon
Robustness
Key nodes
Complex network

ABSTRACT

Cascade phenomenon may lead to catastrophic disasters which extremely imperil the network safety or security in various complex systems such as communication networks, power grids, social networks and so on. In some flow-based networks, the load of failed nodes can be redistributed locally to their neighboring nodes to maximally preserve the traffic oscillations or large-scale cascading failures. However, in such local flow redistribution model, a small set of key nodes attacked subsequently can result in network collapse. Then it is a critical problem to effectively find the set of key nodes in the network. To our best knowledge, this work is the first to study this problem comprehensively. We first introduce the extra capacity for every node to put up with flow fluctuations from neighbors, and two extra capacity distributions including degree based distribution and average distribution are employed. Four heuristic key nodes discovering methods including High-Degree-First (HDF), Low-Degree-First (LDF), Random and Greedy Algorithms (GA) are presented. Extensive simulations are realized in both scale-free networks and random networks. The results show that the greedy algorithm can efficiently find the set of key nodes in both scale-free and random networks. Our work studies network robustness against cascading failures from a very novel perspective, and methods and results are very useful for network robustness evaluations and protections.

1. Introduction

In the modern "network age" [1], various prototypes of networks play more and more important roles in our daily life. Generally, a network is composed of a set of nodes and a set of links connecting the relations between nodes. Every network

* Corresponding authors.
E-mail addresses: zyjiang@xidian.edu.cn (Z.-Y. Jiang), zqliu@jnu.edu.cn (Z.-Q. Liu).

https://doi.org/10.1016/j.physa.2018.02.071
0378-4371/

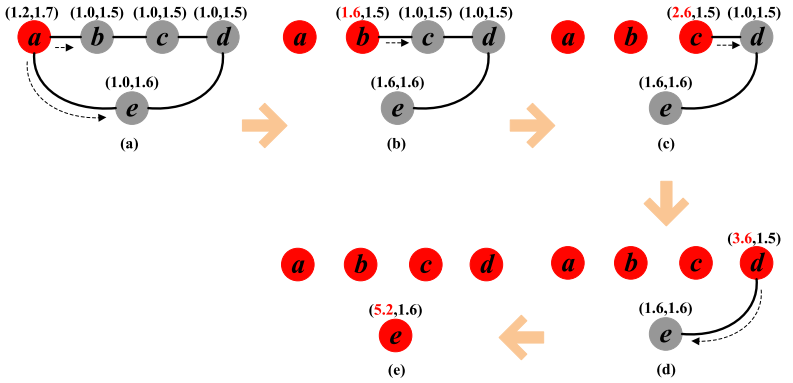

Fig. 1. (Color online). An example for the local flow redistribution and the cascade propagation process. (a) Node a fails, and its flow will be redistributed to its direct neighbors. (b) Assuming both of neighbors b and e receives a half of the load of node a. The load on b exceeds its capacity, and the node b fails. (c) Similarly, the active neighbor c receives the redistributed load from b, and node c fails. In the following process, node d and e fails in subfigures (d) and (e) respectively. Only one node attacked, the entire network collapsed.

has its own unique characteristics and functions. However, there is a common phenomenon called *cascade* that the state or behavior of a node may influence the state or behavior of others in ubiquitous complex networks. For instance, in the power grids, the largest blackout in US history in 2003 [2] resulted in high economic loss. In the ecosystem, the losses of prairie dog in the central US upset the habit in a way that triggered a decline in several animal specials such as owls, bison and so on [3]. In social networks, cross national social influences based on the Facebook [4] triggered a social cascade that fueled the 2010–12 Arab Spring protests [5]. In neuron network, the bursts of small neuronal activity in the brain may lead to neuronal avalanche [6]. In sexually transmitted diseases, a subpopulation of only 2% of the susceptible individuals was responsible for 60% of total infections [7]. In communication networks, traffic congestion on many hub routers may cause the Internet collapse [8]. In the interdependent networks [9,10] and multi-layer networks [11,12], the cascade events can result in unexpected system abnormal. In a word, cascade phenomenon often has catastrophic damage to network and has been a hot research topic in past decades.

So far, from different perspectives, there are many categories of cascading models including *Epidemic Disease* (ED) [13,14] model, *Independent Cascade* (IC) [15] model, *Linear Threshold* (LT) [16] model, *Flow Redistribution* (FR) [17–20] model, and *Percolation-related Model* (PM) [21–23]. In the local flow redistribution model [19], the flow of a failed node was redistributed to its direct neighbors iteratively, resulting in large scale of node failures. It can strongly damage the network function or structure, and deeply imperil the network security or safety [24]. In previous flow redistribution studies [17–19,25–27], a node i has a capacity C_i to deal with the flow L_i of the node. If $L_i \leq C_i$, the node i is in the normal state. Otherwise, the node fails and redistributes its flow to its survived neighbors. The network robustness is evaluated by the size of the giant component against single initial node failure. It is true that there exist one or a few important nodes whose subsequent failures will lead to the network collapse. For network security [24,28,29] purpose, locating the set of these vital nodes is very important for network protection. In the local flow redistribution model [19], it is also very meaningful to find the ***s**et of **k**ey **n**odes* (SKN) whose subsequent failures will result in the whole collapse of a network. In this process, the failure of a node can be divided into two categories, namely *positive failure* and *negative failure*. The positive is triggered by man-made factor such as intentional attacks or self-factor such as random failures by malfunctions, not related to the flow redistribution process. The negative failure is triggered by the flow redistribution here. The nodes in SKN are selected one by one in a sequence. To our best knowledge, nowadays, the problem how to get the SKN in the flow redistribution cascade model is still open.

In a local flow redistribution process, the flow of failed nodes was redistributed locally to their immediate neighbors without global scope. In order to find SKN in which nodes are selected in a sequence, we assume each positive failure is selected from the survival ones after the previous triggered failure cascade propagation terminated, until the whole collapse of the network. By observing the cascading process, it is very interesting that with different vital node sets and failure sequences, the results are very different. For instance, as shown in Fig. 1, the tuple (x, y) near every node represents the flow and capacity of the node respectively. In the steady state, all nodes can operate normally. When a node a fails in Fig. 1(a), assuming its flow 1.2 is equally divided into two parts which are redistributed to its direct neighbors b and e. Then in Fig. 1(b), the flow of b and e is both 1.6 now. The flow of node b exceeds its capacity, and node b fails. Only one survival neighbor is adjacent to node b, and the flow is totally redistributed to node c. Similarly, node c, d, and e fail subsequently. It is a classical cascade propagation process in the local flow redistribution model. However, if assuming the node c fails first in Fig. 2(a), then the two neighbors of node c receive the redistributed flow. No more failures can be triggered by node c. Then we select node a as the second positive failure in Fig. 2(c). The final result can be observed in Fig. 2(d), and node d and e are still survived.

In large scale of complex networks, finding out a SKN of minimum number of nodes quickly is a very challenging problem, because the optimal SKN is a very time consuming process. Therefore, a near optimal SKN discovering method should be

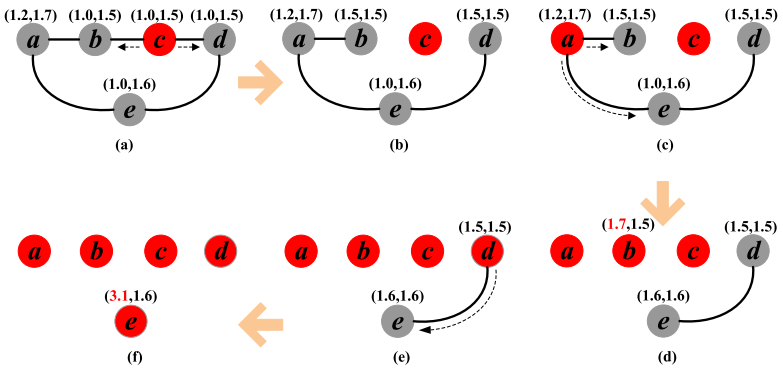

Fig. 2. (Color online). An example for node c, a and d occurring failure subsequently. (a) Node c fails, and the load on c is redistributed to node b and d. (b) No further node fails triggered by node c. (c) Assuming node a fails as the second positive failure, similarly, the load of a is redistributed to b and e. (d) Node b fails. Although two nodes were attacked subsequently, some nodes still survived. (e) Select node d as the third positive failure. (f) Node e fails finally.

deeply studied. In this work, we aim to solve this problem comprehensively. In the following parts, we first introduce models and methods used in this work. Then simulations on classic network models will be discussed, and a conclusion closes this work in the last section.

2. Models & methods

2.1. Capacity models

As discussed in our previous work [18], the traffic flow distributions and node capacity distributions might be very various in different network models. Many empirical studies [30,31] show that the initial flow on every node is strongly related to its degree and can be approximately denoted as

$$L_i = k_i^{\beta}, \tag{1}$$

where β is a parameter and k_i is the degree of node i. $\beta \approx 1.6$ can be observed in many networked systems such as the communication networks. In general, the capacity of a node i should not be less than the flow, and can be notated as

$$C_i = L_i + \varepsilon_i, \tag{2}$$

where ε_i ($\varepsilon_i \geq 0$) is the extra capacity [18] for the flow fluctuations in the network. The distribution of extra capacity of all nodes is very vital to bear the traffic fluctuations in the network. In general, the total extra capacity of all nodes should be a portion of initial total flow of all nodes in the network, and can be defined as

$$W = \sum_i \varepsilon_i = \alpha \sum_i L_i. \tag{3}$$

The parameter α is often used to evaluate the network robustness. In this work, we employ two extra capacity distribution methods. The first is a very classical degree related definition as

$$\varepsilon_i = \alpha L_i. \tag{4}$$

Another simple extra capacity distribution mechanism is the average distribution

$$\varepsilon_i = \frac{W}{N}, \tag{5}$$

where N is the network size (i.e. the number of total nodes in the network).

2.2. Local flow redistribution model

The failure of a selected node may trigger the flow redistribution process, and the redistributed load to a node is related to the flow redistribution model. In the dynamic failure cascading process, the load and active neighbors of every node might change from time to time. Here, we denote the load and the set of non-failed neighbors of any node i at time t as $L_i(t)$ and $\Gamma_i(t)$ respectively. For simplicity, we assume the redistributed flow to a node j (a neighbor of node i) from a failed node i is

$$\Delta L_{i\to j}(t) = L_i(t)\frac{L_j(t)}{\sum_{j'\in\Gamma_i(t)} L_{j'}(t)}, \tag{6}$$

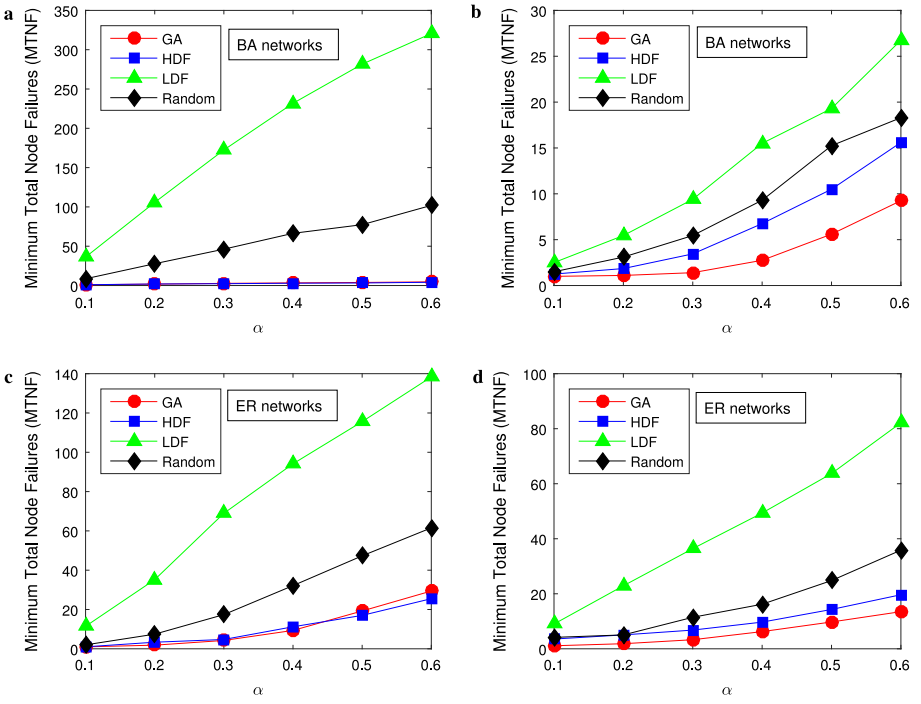

Fig. 3. (Color online). The evolution of MTNF as function of α under the four strategies. (a) In BA scale-free networks using the degree based extra capacity distribution. (b) In BA networks using the average extra capacity distribution. (c) In ER networks using the degree based extra capacity distribution. (d) In ER networks using the average extra capacity distribution. Each datum is the average of more than 20 realizations of networks. Network size $N = 500$, and average degree $\langle k \rangle = 8$.

when the flow redistribution at time t is finished, the flow on node j is now $L_j(t-1) + \Delta L_{i\to j}(t)$. The flow of node j might exceed the capacity and further failure may appear.

2.3. Network models

In this work, we employ two widely used network models including the Barabási–Albert (BA) [32] scale-free network model and Erdös–Rényi (ER) [33] random network model. Many empirical research results have demonstrated that lots of real world networks are associated with scale-free and small-world [34] properties, and we adopt BA [32] network model to represent the infrastructure of complex networks such as the communication networks. The degree distribution of a BA [32] network is $P(k) \sim k^{-3}$. The construction of a BA [32] network is as follows. Starting from m_0 fully connected nodes, a new node with m ($m \leq m_0$) edges is added to the existing network, and the other end of every new edge is selected preferentially according to the probability

$$\Pi_i = \frac{k_i}{\sum_j k_j}, \tag{7}$$

where k_i and k_j are the degrees of node i and j respectively.

The generation of a ER [33] random network is simple and efficient. Beginning with N isolated nodes, a link is connected between every pair of nodes with probability p. Finally, a random network of about $N(N-1)/2$ undirected links is composed.

2.4. SKN discovering methods

In a network, a node of the highest degree is often considered to be important and has high effects on network performance. [35,36] Then we propose a high degree first mechanism for SKN, and we denote the number of nodes in SKN as the ***m**inimum **t**otal **n**ode **f**ailures* (MTNF).

Method 1: **H**igh-**D**egree-**F**irst (HDF)

Step 1: set MTNF = {};

Step 2: find the survived node of the highest degree in the current network, denoted as i, MTNF = {MTNF, i};

Step 3: set the state of i to be failed, and trigger the flow redistribution process, until no further failure propagation;

Step 4: if all nodes in the network are failed, the process is finished; otherwise, go to step 2.

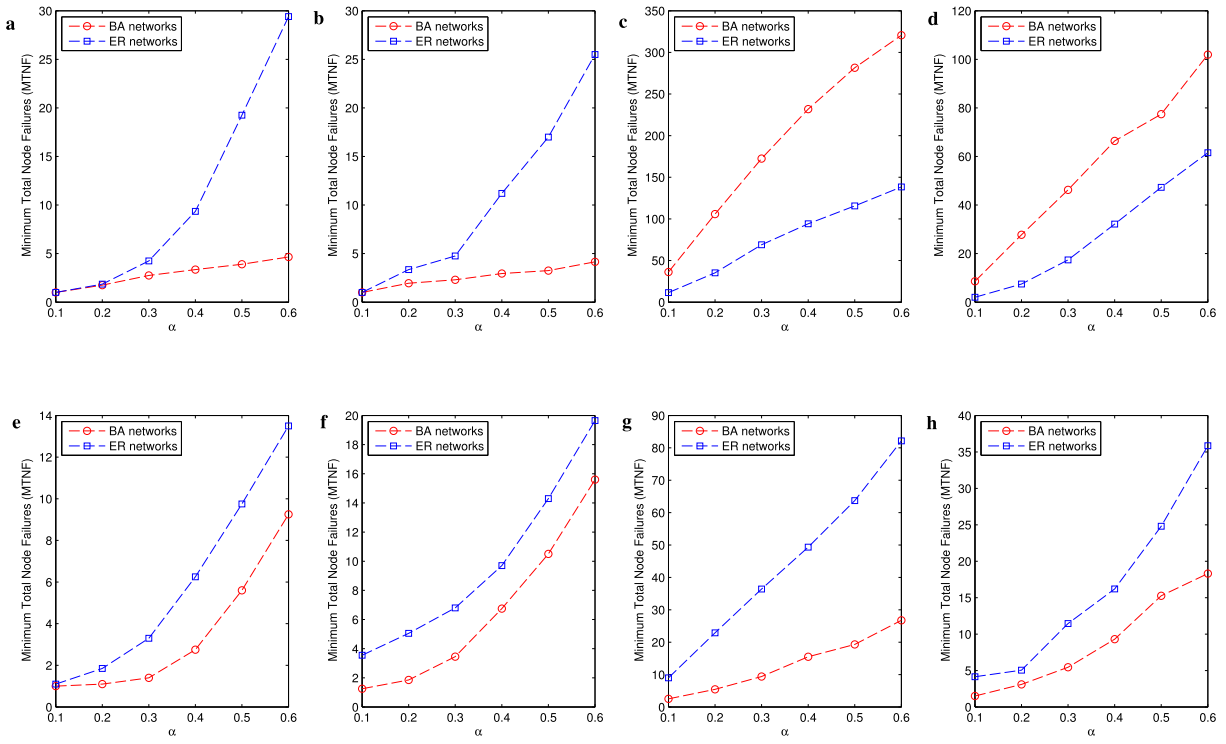

Fig. 4. (Color online). Comparisons of MTNF in different networks under the all extra capacity distributions and strategies. (a) Degree based extra capacity distribution, under the GA. (b) Degree based extra capacity distribution, under the HDF. (c) Degree based extra capacity distribution, under the LDF. (d) Degree based extra capacity distribution, under the random method. (e) Average extra capacity distribution, under the GA. (f) Average extra capacity distribution, under the HDF. (g) Average extra capacity distribution, under the LDF. (h) Average extra capacity distribution, under the random method. Each datum is the average of more than 20 realizations of networks. Network size $N = 500$, and average degree $\langle k \rangle = 8$.

Meanwhile, in order to compare with HDF, the low degree first method is often used and described as follows.

Method 2: **L**ow-**D**egree-**F**irst (LDF)

Step 1: set MTNF = {};

Step 2: find the survived node of the lowest degree in the current network, denoted as i, MTNF = {MTNF, i};

Step 3: set the state of i to be failed, and trigger the flow redistribution process, until no further failure propagation;

Step 4: if all nodes in the network are failed, the process is finished; otherwise, go to step 2.

Without degree information, the random selection method is the simplest, and here is employed as the benchmark of other methods.

Method 3: Random

Step 1: set MTNF = {};

Step 2: find a survived node randomly in the current network, denoted as i, MTNF = {MTNF, i};

Step 3: set the state of i to be failed, and trigger the flow redistribution process, until no further failure propagation;

Step 4: if all nodes in the network are failed, the process is finished; otherwise, go to step 2.

Inspired by our previous work [28,35], at each step, based on the existing network, the number of total failed node triggered by every survived node can be obtained, and we can choose the one which can lead to the largest failure scale of survived nodes.

Method 4: **G**reedy **A**lgorithm (GA)

Step 1: set MTNF = {};

Step 2: find a survived node that can lead to the largest failure scale in the current survived network, denoted as i, MTNF = {MTNF, i};

Step 3: set the state of i to be failed, and trigger the flow redistribution process, until no further failure propagation;

Step 4: if all nodes in the network are failed, the process is finished; otherwise, go to step 2.

3. Simulations

We first evaluate the evolution of MTNF as a function of α in Eq. (2) under different SKN discovering methods in two classical network models.

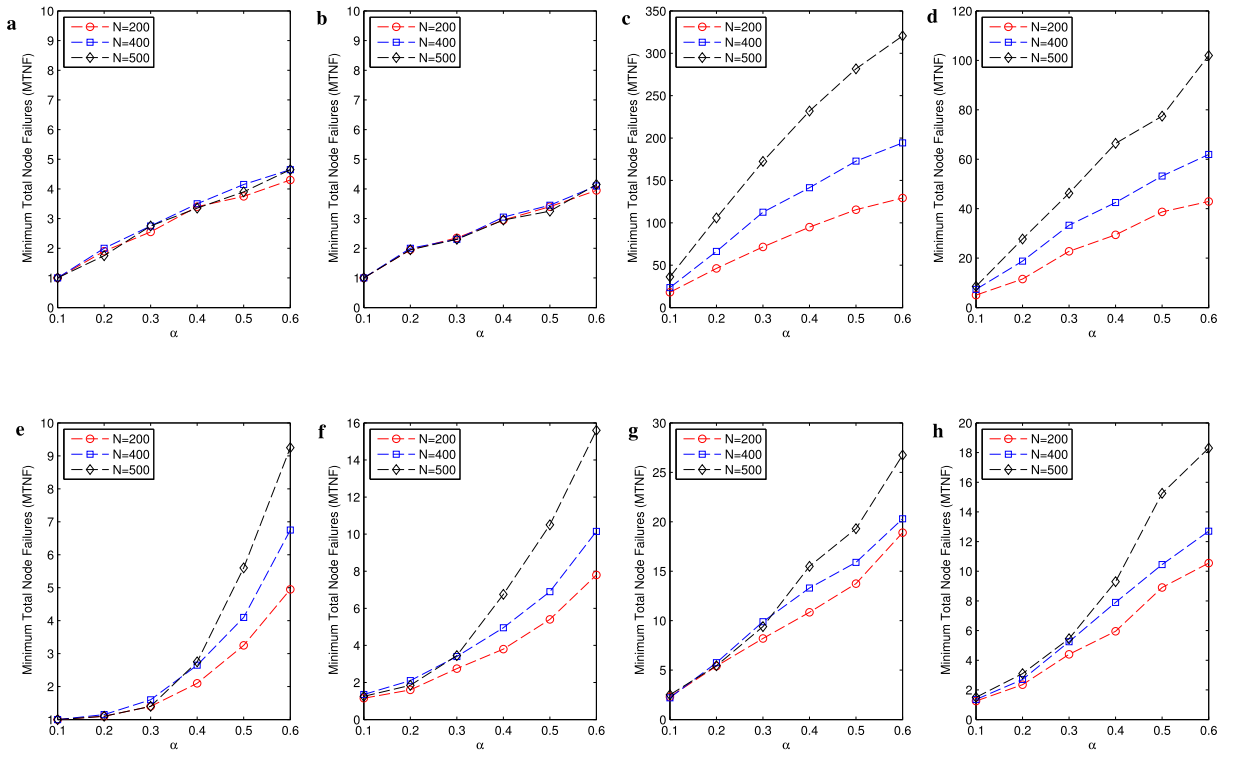

Fig. 5. (Color online). Comparisons of MTNF for different network sizes in BA [32] networks. (a) Degree based extra capacity distribution, under the GA. (b) Degree based extra capacity distribution, under the HDF. (c) Degree based extra capacity distribution, under the LDF. (d) Degree based extra capacity distribution, under the random method. (e) Average extra capacity distribution, under the GA. (f) Average extra capacity distribution, under the HDF. (g) Average extra capacity distribution, under the LDF. (h) Average extra capacity distribution, under the random method. Each datum is the average of more than 20 realizations of networks.

In Fig. 3(a), based on degree based extra capacity distribution, in BA [32] scale-free networks, the LDF is the worst, and the MTNFs of the HDF and greedy algorithm are almost the same. In Fig. 3(b), based on the average extra capacity distribution in BA [32] scale-free networks, the greedy algorithm achieves the best results for all network sizes in simulations. In ER networks, in Fig. 3(c), using the degree based extra capacity distribution, it is interesting that the greedy algorithm has lower MTNF than HDF method does when $\alpha \leq 0.4$ in our simulations, but higher than HDF when $\alpha > 0.4$. But the results are very close. In Fig. 3(d), based on the average extra capacity distribution, the results are very similar to the results in Fig. 3(b). With increasing α, the MTNF increases. Because the higher total extra capacity the network has, the higher robustness the network is. On the whole, compared with the other methods, the greedy algorithm has the best results for different extra capacity distributions and in different network models.

Based on the same extra capacity distribution, and under the same SKN discovering method, the MTNFs might be very different in different network models. In Fig. 4(a)–(d), based on the degree based extra capacity distribution, one can see that under the greedy algorithm and HDF, the MTNF of BA [32] networks is smaller than that of ER [33] networks. Meanwhile, under the LDF and random selection, the MTNF of ER [33] networks appear to be better than that of BA [32] networks. This mainly because the degree based extra capacity distribution is strongly related to the node degrees, and the degree centrality is more obvious in BA [32] networks. In Fig. 4(e)–(h), based on the average extra capacity distribution, the results of BA [32] networks seem to be better than that of ER [33] networks under the four SKN discovering methods.

In general, with increasing network size N, the network structure becomes more complex. In Figs. 5 and 6, we evaluate the effects of network sizes on MTNF in BA [32] networks and ER [33] networks respectively. It should be noticed that based on the degree based extra capacity distribution, under the greedy algorithm and HDF, the MTNFs are almost the same under the three network sizes (200 400 500) in BA [32] networks as shown in Fig. 5(a)–(b). Meanwhile, in ER [33] networks, the MTNF appears to increase with network size. Under the LDF and random method, and based on the both extra capacity distributions, the MTNF increases with increasing network sizes and α. In BA [32] networks, due to the degree centrality, the nodes with high degrees often influence the whole network, the greedy algorithm and HDF preferentially select these nodes of high degree, and then the MTNF appears to keep steady for different network sizes. From another perspective, the greedy algorithm and HDF can efficiently discover the SKN of BA [32] scale-free networks.

In our opinion, MTNF is a very important network performance metric in network science research for the network robustness evaluations. The above mentioned methods are all considered to be heuristic ones. The results are not optimal.

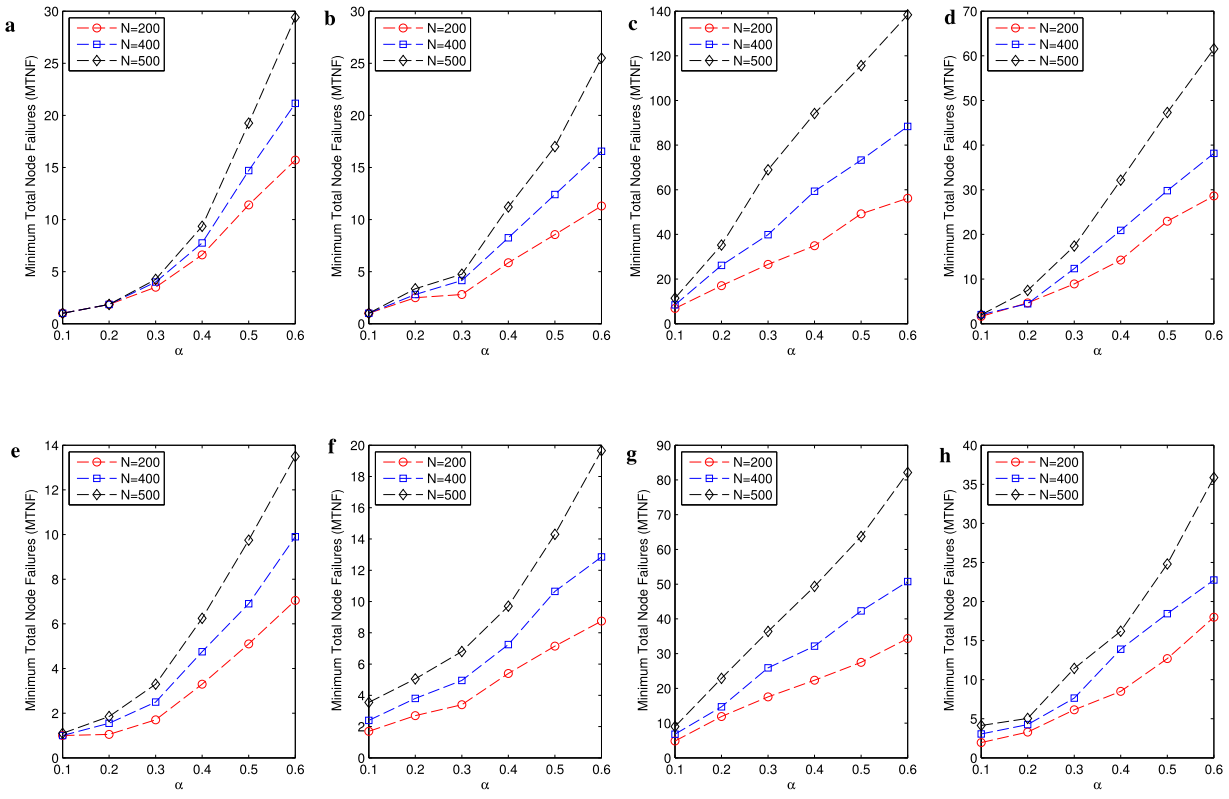

Fig. 6. (Color online). Comparisons of MTNF for different network sizes in ER [33] networks. (a) Degree based extra capacity distribution, under the GA. (b) Degree based extra capacity distribution, under the HDF. (c) Degree based extra capacity distribution, under the LDF. (d) Degree based extra capacity distribution, under the random method. (e) Average extra capacity distribution, under the GA. (f) Average extra capacity distribution, under the HDF. (g) Average extra capacity distribution, under the LDF. (h) Average extra capacity distribution, under the random method. Each datum is the average of more than 20 realizations of networks.

To our best knowledge, there is only one way to obtain the optimal results, namely exhaustive search which tries all possible sequences of nodes. In this work, in order to compare with the optimal results, we set the network size $N = 60$. As shown in Fig. 7(a)–(b), the greedy algorithm can achieve near optimal results in BA [32] scale-free networks under degree based extra capacity distribution and average extra capacity distribution. In Fig. 7(c), in ER [33] networks, with the degree based extra capacity distribution, the HDF method appears to be close to the optimum. While in Fig. 7(d), the GA achieves better network capacity than any other three heuristic methods.

Degree distribution of most empirical networks such communication networks, social networks, highway networks appears to be power-law, and the HDF and GA methods are very suitable to discover the SKN sets for these real networked systems. Therefore, our work has potential applications to find the network reliability bottlenecks against subsequent attacks. Furthermore, are there any effective protection mechanisms to enhance the network reliability against such node-group-based attacks?

4. Conclusions

Failure cascading phenomenon can lead to very serious network security disasters. This work mainly concerned on one kind of cascade propagation problems. The flow redistribution process was dynamic, and subsequent node positive failures may occur and result in the network collapse. Thus the set of key nodes whose subsequent failures would lead to the network collapse was very vital and should be efficiently discovered. Inspired by many previous studies, in this work, we proposed four SKN discovering methods including High-Degree-First, Low-Degree-First, Random and Greedy Algorithm. We composed extensive simulations and discussed the related results which showed that the greedy algorithm and high-degree-first methods can efficiently find the SKNs for BA scale-free network models. Moreover, we compared them with the optimal results to further verify the effectiveness of the proposed methods. This work studied the cascade phenomenon from a new perspective, and our work presented four heuristic SKN researching methods. In addition, node-group-based attacks have higher destructiveness to network security, and how to effectively protect such group-based attacks still appears to be an open problem.

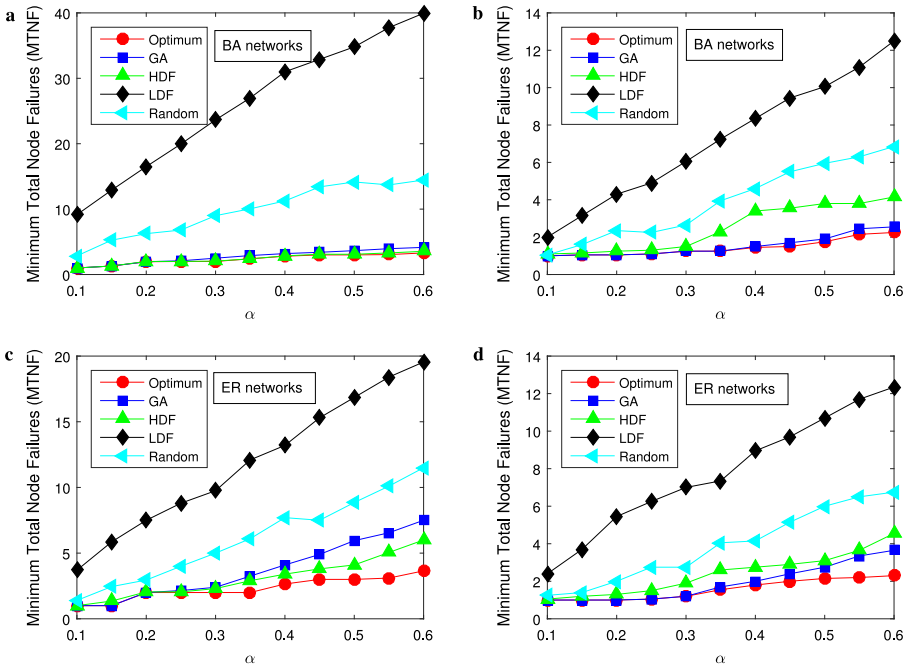

Fig. 7. (Color online). Comparisons of MTNF results under different methods. (a) In BA scale-free networks using the degree based extra capacity distribution. (b) In BA networks using the average extra capacity distribution. (c) In ER networks using the degree based extra capacity distribution. (d) In ER networks using the average extra capacity distribution. Each datum is the average of more than 20 realizations of networks. Network size $N = 60$, and average degree $\langle k \rangle = 8$.

Acknowledgments

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions. This work was partly supported by the National Natural Science Foundation of China (No. 61502375), the Natural Science Basis Research Plan in Shaanxi Province of China (No. 2016JQ6046), the Fundamental Research Funds for the Central Universities, China (No. JB171502), the Key Program of NSFC-Guangdong Union Foundation, China (No. U1405255), the National High Technology Research and Development Program (863 Program), China (No. 2015AA016007 and 2015AA017203), and the China 111 Project (No. B16037).

References

[1] A.E. Motter, Y. Yang, Phys. Today 70 (1) (2017) 32.
[2] J. Glanz, R. Perez-Pena, 90 seconds that left tens of millions of people in the dark, New York Times (2003).
[3] B.J. Bergstrom, L.C. Arias, A.D. Davidson, et al., Conserv. Lett. 7 (2) (2014) 131–142.
[4] www.facebook.com.
[5] C.D. Brummitt, G. Barnett, R.M. DSouza, J. R. Soc. Interface 12 (112) (2015) 20150712.
[6] J.M. Palva, A. Zhigalov, J. Hirvonen, et al., Proc. Natl. Acad. Sci. USA 110 (9) (2013) 3585–3590.
[7] J.A. Yorke, H.W. Hethcote, A. Nold, Sex. Transm. Dis. 5 (2) (1978) 51–56.
[8] V. Jacobson, Comput. Commun. Rev. 18 (1998) 314.
[9] Z. Chen, J. Wu, Y. Xia, X. Zhang, Robustness of interdependent power grids and communication networks: A complex network perspective, IEEE Trans. Circuits Syst. II: Express Briefs 99 (2017) 1.
[10] F. Tan, Y. Xia, Z. Wei, Phys. Rev. E 91 (2015) 052809.
[11] W.B. Du, X.L. Zhou, O. Lordan, Z. Wang, C. Zhao, Y.B. Zhu, Analysis of the Chinese Airline Network as multi-layer networks, Transp. Res. Part E: Logist. Transp. Rev. 89 (2016) 108–116.
[12] W.B. Du, Y. Gao, C. Liu, Z. Zheng, Z. Wang, Adequate is better: particle swarm optimization with limited-information, Appl. Math. Comput. 268 (2015) 832–838.
[13] X. Fu, M. Small, G. Chen, John Wiley & Sons, 2013.
[14] M. Gomez Rodriguez, J. Leskovec, A. Krause, Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 1019–1028.
[15] J. Goldenberg, B. Libai, E. Muller, Mark. Lett. 12 (3) (2001) 211–223.
[16] M. Granovetter, Am. J. Sociol. 1978 (1978) 1420–1443.
[17] A.E. Motter, Phys. Rev. Lett. 93 (2004) 098701.
[18] Z.Y. Jiang, J.F. Ma, Internat. J. Modern Phys. C 28 (2017) 1750031.
[19] J.-W. Wang, L.-L. Rong, Saf. Sci. 47 (2009) 1332.
[20] Z.-Y. Jiang, J.-F. Ma, Y.-L. Shen, Y. Zeng, Physica A 457 (2016) 1–7.

[21] S.V. Buldyrev, R. Parshani, G. Paul, et al., Nature 464 (7291) (2010) 1025–1028.
[22] R.R. Liu, M. Li, C.X. Jia, Sci. Rep. 6 (2016) 35352.
[23] R. Parshani, S. Buldyrev, S. Havlin, Phys. Rev. Lett. 105 (2010) 048701.
[24] C. Shen, H.G. Zhang, D. Feng, et al., Sci. China Inf. Sci. 50 (3) (2007) 273–298.
[25] A.E. Motter, Y.-C. Lai, Phys. Rev. E 66 (2002) 065102.
[26] X.-B. Cao, C. Hong, W.-B. Du, J. Zhang, Chaos Solitons Fractals 57 (2013) 35.
[27] J. Wang, Saf. Sci. 53 (2013) 219–225.
[28] J. Liu, Z. Jiang, N. Kato, et al., IEEE Wirel. Commun. 23 (3) (2016) 90–96.
[29] C. Pu, S. Li, A. Michaelson, J. Yang, Iterative path attacks on networks, Phys. Lett. A 379 (2015) 1633–1638.
[30] K.-I. Goh, B. Kahng, D. Kim, Phys. Rev. Lett. 87 (2001) 278701.
[31] K. Park, Y.-C. Lai, N. Ye, Phys. Rev. E 70 (2004) 026109.
[32] A.-L. Barabási, R. Albert, Science 286 (1999) 509.
[33] P. Erdös, A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. 5 (1960) 17.
[34] D.J. Watts, S.H. Strogatz, Nature 393 (1998) 440.
[35] Z.Y. Jiang, J.F. Ma, Sci. Rep. 7 (2017) 40428.
[36] W. Du, B. Liang, G. Yan, O. Lordan, X. Cao, Chin. J. Aeronaut. 30 (2017) 330–336.