Understanding and preventing cascading breakdown in complex clustered networks

Liang Huang,[1] Ying-Cheng Lai,[1,2] and Guanrong Chen[3]
[1]*Department of Electrical Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2]*Department of Physics and Astronomy, Arizona State University, Tempe, Arizona 85287, USA*
[3]*Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China*
(Received 17 April 2008; revised manuscript received 13 August 2008; published 30 September 2008)

Complex clustered networks are ubiquitous in natural and technological systems. Understanding the physics of the security of such networks in response to attacks is of significant value. We develop a model, based on physical analysis and numerical computations, for the key ingredients of load dynamics in typical clustered networks. With this understanding, an effective strategy is proposed for preventing cascading breakdown, one of the most disastrous events that can happen to a complex networked system.

DOI: 10.1103/PhysRevE.78.036116 PACS number(s): 89.75.Hc, 89.20.Hh, 05.10.−a

Recently, cascading breakdown [1–3] in complex networks has received considerable attention [4–7]. The phenomenon is referred to as an avalanching type of process, where the failure of a single or of a few nodes can result in a large-scale breakdown of the network. In particular, in a physical network nodes carry and process certain loads, such as electrical power, and their load-bearing capacities are finite. When a node fails, the load that it carries will be redistributed to other nodes, potentially triggering more failures in the network as a result of overloading. This process can propagate through the entire network, leading to its breakdown. Indeed, cascading breakdown appears to be particularly relevant for large-scale failures of electrical power grids, and efforts have been made to understand the dynamical origin of such failures [8]. From the standpoint of network security, scale-free networks [9], where a small subset of nodes (hubs) possess substantially more links than those of an average node and therefore carry disproportionally more loads, are especially vulnerable to cascading breakdown, as attack on one of the hub nodes can cause a significant load redistribution [2,5]. In this regard, a strategy for protecting scale-free networks against cascading breakdown has been proposed [6], where a selective set of "unimportant" nodes that process little but contribute relatively large loads to the network are pre-emptively removed so as to reduce the overall load in the network.

Networks with a community structure, or clustered networks, are relevant to a plethora of biological, social, and technological systems [10]. A clustered network consists of a number of groups, where nodes within each group are densely connected but the linkage among the groups is sparse. A clustered network can be heterogeneous in the sense that its degrees obey a power-law distribution, which can be realized, for example, by incorporating the scale-free topology in each cluster. Recently various dynamics on complex clustered networks have been studied [11].

In this paper, we address the dynamical origin of cascading processes on complex clustered networks and, more importantly, investigate how such a network can be made secure in response to attacks. In view of the particular vulnerability of scale-free networks to cascading breakdown, we focus on networks where each individual cluster contains a scale-free subnetwork. To motivate our work and illustrate the challenges, we consider the problem of virus spread starting from one of the clusters, such as a remote village in a human epidemic network. A common practice to prevent a global spread is to isolate this particular cluster from the network. Now, consider the network-security problem by assuming that an attack has occurred in one of the clusters. A naive strategy to prevent breakdown of the network on a global scale is to isolate this cluster by cutting all the links that connect this cluster with other clusters so that failures would be restricted to the original cluster. This intuitive thinking, however, cannot be correct for a load-distributed network, because cutting off a cluster would transfer the load originally processed by this cluster to other clusters of the network, increasing the likelihood of overloading and possibly resulting in a more disastrous situation. Indeed, this is what we have found in simulations: a clustered network is particularly vulnerable to cascading breakdown in the sense that the general prevention strategy in Ref. [6], which is quite effective for scale-free networks, would increase significantly the probability of a global avalanche if not properly implemented.

Our main idea is to classify and understand the roles played by various nodes in the network and devise a control strategy accordingly that can effectively prevent global cascades. Our achievement is illustrated in Fig. 1, plots of the relative size G of the largest connected component of the network versus some generic network capacity parameter λ in response to an attack on a hub node, where $G=1$ represents a fully connected network and $G \ll 1$ indicates that the network has been disintegrated effectively. The data points represented by open squares correspond to the situation where no control is taken to protect the network, and those represented by open circles are the result of cutting off the particular cluster within which the attack occurs. We observe that, as the capacity parameter λ is reduced, G decreases rapidly but strikingly, there is essentially no difference in the values of G between these two cases, indicating the ineffectiveness of an straightforward implementation of the prevention strategy which tries to localize the destruction within one community. In contrast, implementing our control strategy results in much higher values of G (data points represented by open triangles). In what follows, we present a sequence of reasonings, supported by numerical computations, that lead to a relatively complete understanding of the cascading phenomenon in complex clustered networks and, consequently, to an effective control strategy.

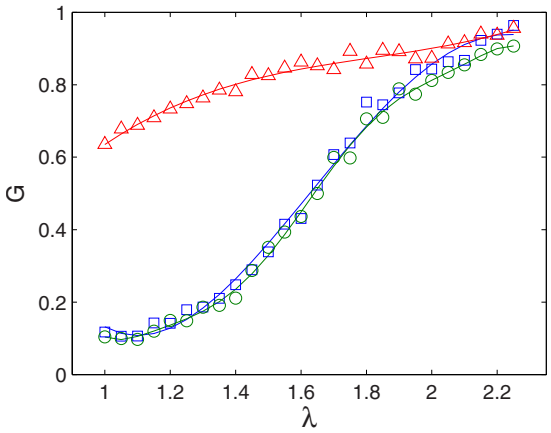

FIG. 1. (Color online) For a representative clustered network of $N=5600$ nodes, average degree $\langle k\rangle=4$, $M=50$ clusters, and average number of intercluster links $k_M=2$, the relative size G of the largest connected component in the network versus the network capacity parameter λ in response to a targeted attack. Each data point is the result of averaging 100 network realizations (see text for details of the meanings of the three different data curves). The attack disables a single node that has the largest load. For a nonclustered scale-free network, the value of G can be about zero for $\lambda \simeq 1$ [2]. However, for a clustered network, failures propagate from one cluster to another, during which a few connected clusters may be separated from the rest but still remain connected. As a result, the value of G for small values of λ is small but not zero; it is of the order of $1/M$.

We consider an ensemble of clustered networks, each of N nodes and M clusters, where $N \gg M$. Any cluster within the network is a scale-free subnetwork of $n=N/M \gg 1$ nodes [9]. The number of intercluster links is $k_M M$, and they are placed randomly among the clusters. To conserve the average degree of the entire network, we cut off $k_M M$ intracluster links randomly while keeping the network fully connected. Since the number of intercluster links is much smaller than that of intracluster links, removing a small number of intracluster links has little effect on the dynamics of the network. To investigate cascading breakdown, we use the prototypical model of load dynamics [2]. In particular, the load L_i at node i is defined as the total number of directed shortest paths passing through this node. Paths that end at or start from the node are also counted. The total load of the network is given by $S=\Sigma L_i=N(N-1)(D+1)$, where D is the average network distance. The capacity of a node is the maximum load that the node can handle, which is assumed to be proportional to its initial load L_{i0}: $C_i=\lambda L_{i0}$, where the constant $\lambda \geqslant 1$ is a uniform capacity parameter. An attack at a particular node is defined as an event that disables or removes this node from the network. If the load that this node handles is relatively large, a load redistribution over the network can occur. Any node in the network is considered to have failed and is removed from the network if the load imposed on it is larger than its capacity. The damage after the network reaches a new steady state can be conveniently quantified by the relative size $G=N'/N$, where N' is the number of nodes in the largest connected component remaining after the attack. For $G \lesssim 1$, the network remains mostly connected, so the effect of attack on the network is not severe. For $G \gtrsim 0$, breakdown of the network occurs at a global scale.

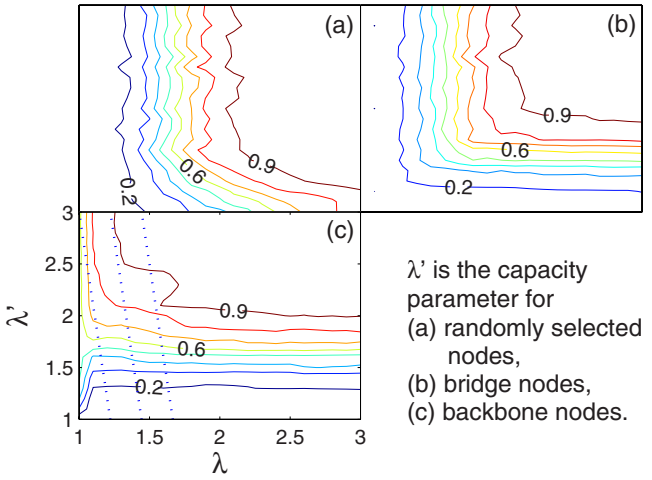

FIG. 2. (Color online) Contour plots of G versus λ and λ'. The dotted lines in (c) correspond to effective total capacity parameter of 1.2, 1.4, and 1.6 (from left to right). Network and simulation parameters are the same as for Fig. 1.

To understand the dynamical origin of cascading failures in a clustered network, we note that nodes connecting different clusters, or *bridge nodes*, transmit intercluster load flows and they are critical to maintaining the global connections of the network. For the ensemble of networks used in Fig. 1, we find that the fraction of the bridge nodes is about 3.5%, but they carry about 41% of the total load of the network. An intuition is, then, that assigning relatively large capacities to the bridge nodes may mitigate cascades. To test this hypothesis, we conduct the following numerical experiments. First, we randomly select a set of nodes, which has the same number as that of the bridge nodes, and assign them with different capacities as characterized by the parameter λ' (the remaining nodes in the network have the capacity parameter λ). We then examine, in the two-dimensional parameter plane (λ,λ'), contours of various values of G. The result is shown in Fig. 2(a), where the contours are mostly vertical, indicating little dependence of G on λ'. Thus, having a random set of nodes with high capacities cannot help prevent cascading failures, as expected. Next, we assign λ' but only to the set of bridge nodes. As shown in Fig. 2(b), in this case, the contour lines are approximately symmetric with respect to $\lambda'=\lambda$, indicating that G depends mainly on λ' but only in the region where $\lambda'<\lambda$. For $\lambda'>\lambda$, G has little dependence on λ', revealing the ineffectiveness of having high-capacity bridge nodes in limiting cascading failures. There is in fact a bottleneck effect at the bridge nodes: if their capacities are small, they will hinder the load-transferring capability of the network, but increasing their capacities in general can only facilitate load transfers among the clusters via intercluster links. Since the majority of links in the network are intracluster links, load transfers within individual clusters are prevalent. As a result, having large-capacity bridge nodes cannot enhance the network's load-transferring ability in general.

The results in Figs. 2(a) and 2(b) suggest the need to identify a different set of nodes that are more important to the load dynamics than the bridge nodes. Our key idea is to

examine, within any given cluster, the set of nodes that are on the shortest paths connecting the bridge nodes. We call such nodes *skeleton nodes*, as the shortest paths through them are the main avenues for load transfers within the cluster. The bridge and the skeleton nodes thus form the *backbone* of load traffic on the network. Indeed, for the model network in Fig. 1, the fraction of these two types of backbone nodes is 13% but they carry 79% of the total load. A typical scenario for traffic flow on the network is then as follows. Say node A in one cluster wishes to transfer a certain amount of load to node B in a different cluster. Node A first sends the load to a closest skeleton node in the same cluster, which will then be sent to a bridge node along the shortest path. Such shortest paths can be regarded as "highways" for load traffic. The load is then transported to the destination cluster along a series of "highways" connecting various backbone nodes. Upon arrival at the destination cluster, the load is finally sent to node B via some "local" connections in that cluster. This picture is analogous to the surface transportation system in a modern infrastructure. We may expect that increasing the capacities of the backbone nodes can reduce the likelihood of overloading in the network, thereby making the network more tolerant to cascading breakdown. Figure 2(c) shows the contours of a number of values of G in the (λ,λ') plane, where λ' now is the capacity parameter for both types of backbone nodes. Indeed, for a fixed value of λ, as λ' is increased, G can be increased significantly. Setting a high value of λ' is practical, as the number of backbone nodes is small (typically about 10% of the total number of nodes in the network). To give a concrete example, assume first all nodes have the same capacity: $\lambda'=\lambda=1.4$. After the attack, G is about 0.3, indicating that only 30% of the nodes are still connected. However, if we set $\lambda'=2.3$ and $\lambda=1.3$ so that the total capacity of the network is the same as for the case of $\lambda'=\lambda=1.4$, we find that G can be maintained at about 0.9, a three-fold increase over the previous case.

The above analysis suggests an effective way to implement the strategy of removing "unimportant" nodes in the network to prevent cascading breakdown [6], i.e., to remove a certain fraction of *nonessential nodes* that are neither skeleton nor bridge nodes. These nonessential nodes contribute loads to the network but they process or transfer little loads, so a controlled removal can reduce the total load while keeping intact the overall traffic flow of the network. A key issue is the optimal fraction of the nonessential nodes that should be removed to maximize the network's robustness against cascading breakdown. In the following, we develop a physical analysis and numerical computations to address this issue.

We order the clusters by their average distances to the cluster under attack. In particular, we denote the cluster where a cascading process is originated as $I_M=1$ and calculate the average distances between nodes in this cluster and nodes in other clusters: $l_{1J}=1/n^2\Sigma d_{ij}$, $J=2,3,\ldots,M$, where the sum is over all nodes i in cluster 1 and all nodes j in cluster J. The average distances l_{1J} are arranged in an ascending order, i.e., the cluster that has the smallest distance l_{1J} is denoted by $I_M=2$, and so on. The order thus characterizes the closeness of an arbitrary cluster to the cluster under attack. We find that removing nonessential nodes from clusters that are close to the original cluster can lead to higher values of G, as shown in Fig. 3(a). This can be understood as follows. By removing some nonessential nodes in a cluster, the load decrease in the skeleton and bridge nodes in this cluster is $n_n(N-n)d$, where $n_n \sim n$ is the number of nonessential nodes, and d is the average path length for load at a nonessential node to travel through the backbone nodes in this cluster. The load decrease over all backbone nodes is approximately $n(N-n)(D-d)$. Because of the clustered topology of the network, D is much larger than d. For example, for the parameters used in Fig. 2, $D \simeq 14$ and $d \simeq 2$. The average load decrease associated with the backbone nodes in each cluster is then $n(N-n)(D-d)/(M-1)$, which is much less than the load decrease in the original cluster. In general, the closer a cluster is to the original cluster, the more load decrease occurs. Thus, to significantly increase the network's ability to resist cascading breakdown while at the same time to minimize its impact on the network, nonessential nodes in clusters that are closer to the original cluster should be targeted for removal. Figure 3(b) shows this effect by comparing the consequence of removing nonessential nodes from randomly selected clusters and from clusters that are more distant from the original cluster. We see that removing nonessential nodes from close clusters results in about 20% of improvement in G as compared with node removal from ran-

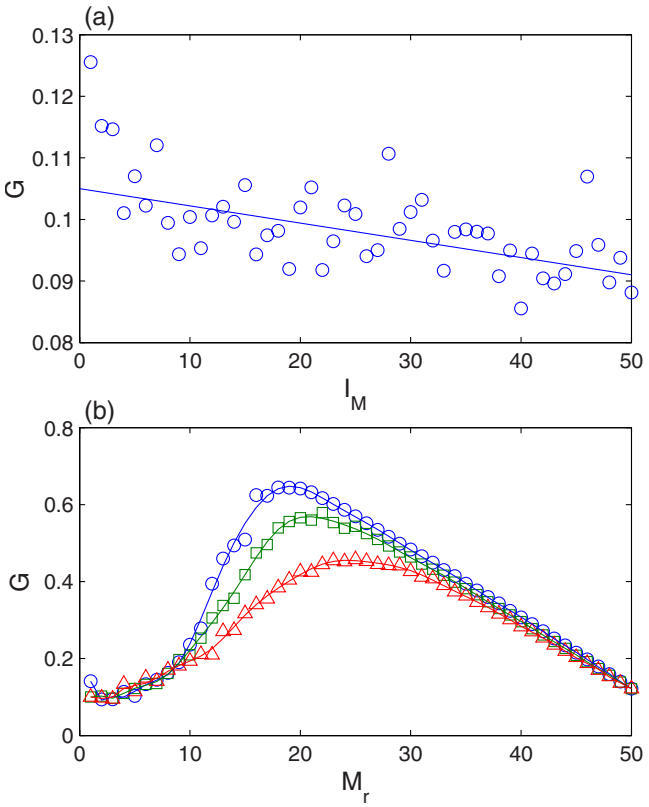

FIG. 3. (Color online) For $\lambda=1$, (a) G versus the index I_M of the cluster from which nonessential nodes are removed. (b) G versus the number of clusters M_r where controlled removal occurs. Circles: from clusters with small index to large index; squares: from randomly selected clusters; triangles: from clusters with large index to small index.

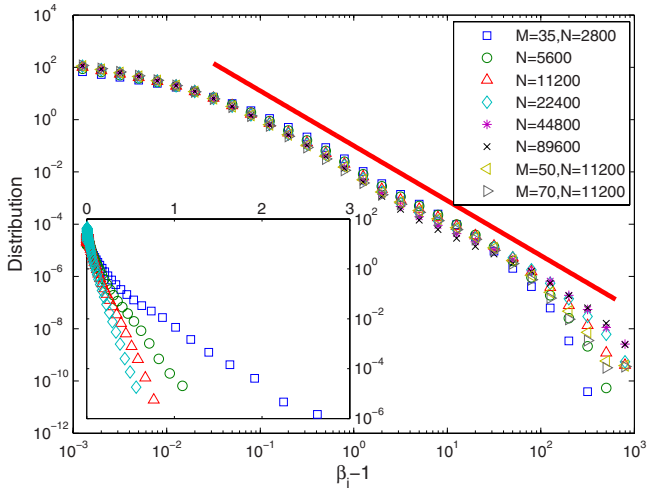

FIG. 4. (Color online) The distribution of the shifted load ratio β_i-1. For clustered networks, the distribution has a long tail and is independent to network details such as the number of clusters and network size, indicating the existence of large load fluctuations before and after attack. The straight line has a slope of −2.1. Inset: the same quantity for a single scale-free network (by setting $M=1$). $N=5600$, $\langle k\rangle=4,6,8,10$ from right to left. The distribution for the shifted ratio is exponential. Each data is the result of averaging at least 100 random realizations.

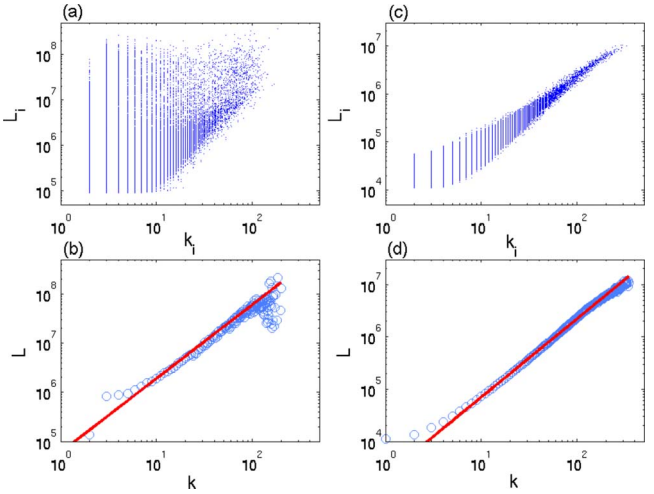

FIG. 5. (Color online) Load versus degree for clustered networks with $M=35$, $N=44800$ (a), (b), and for scale-free networks with $N=5600$, $\langle k\rangle=4$ (c), (d). (a), (c) Scattered plots for all (k_i,L_i) pairs. (b), (d) The averaged load L over all the nodes with the same degree versus node degree k. The straight line has a slop of 1.5. The data are obtained from more than 100 random realizations.

domly chosen clusters, and the improvement is about 50% when comparing with removal from some more distant clusters.

For controlled node removal from some randomly chosen clusters, the optimal removing size M_{RC} that maximizes G can be estimated, as follows. Before removal, the total load is $S=N(N-1)(D+1)$. After removing int$[fN]$ nonessential nodes, the total load becomes $S'=N'(N'-1)(D'+1)$, where $N'=\text{int}[(1-f)N]$ and D' is the new network distance. Since the backbone nodes play a dominant role in load processing, $D'\approx D$ and $S'/S\approx(1-f)^2$. That is, the load of an average backbone node i decreases by a factor of $(1-f)^2$ as the result of controlled removal. After the attack, the load of node i will in general increase from L_i to $L_i'=\beta L_i$, where β is a constant depending on the network structures. The new load can thus be written as $(1-f)^2\beta L_i$. If the capacity λL_i of node i is larger than the new load, i.e., $\lambda L_i>(1-f)^2\beta L_i$, cascading failures will not occur. In this sense, the quantity β characterizes the network's ability to resist cascading breakdown.

Generally, the value of the parameter β depends on nodes, thus it is more accurate to write $L_i'=\beta_i L_i$. Most of the nodes in the network have β values close to 1, with a small set of nodes having larger β values. The probability distribution of β_i-1 is shown in Fig. 4. We observe that for scale-free networks without clustered structure, the distribution decays exponentially for large β_i (inset of Fig. 4). This is consistent with previous results that for networks without a clustered structure, $\beta-1\approx 0$ [5]. However, for a clustered network, the distribution of β_i has a long tail compared with exponential decay, indicating large load fluctuations after the initial attack. Heuristically, this could be understood, as follows. A single network is compact and its structure is homogeneous, i.e., removing some nodes results in a smaller network but with similar properties. For example, for a scale-free network the load L_i and the degree k_i satisfy the scaling relation $L_i\sim k_i^\alpha$, where $\alpha\approx 1.5$. After removing a few nodes, it is still a scale-free network, thus the relation $L_i'\sim k_i'^\alpha$ still holds, where prime means the corresponding network quantities after the removal. Since the number of nodes removed is small, one expects the change in the degree to be small as well, thus $k_i'\approx k_i$, and $L_i'\approx L_i$ [5]. However, for a clustered network, although the averaged load $L(k)$ over the nodes with the same degree k scales as $L\sim k^\alpha$ [Fig. 5(b)], the relation does not hold for individual nodes [Fig. 5(a)], contrasting with that of scale-free networks [Fig. 5(c)]. Indeed, the load for such a network is determined by the type of the nodes. Generally, the bridge nodes have the largest loads, followed by the skeleton nodes, and then by the nonessential nodes. Since the links between clusters are established among randomly selected nodes, the backbone nodes can have both large and small degrees [Fig. 5(a)]. Furthermore, when the network is attacked, the backbone structure is altered. On one hand, some new nodes may become backbone nodes, and their loads will increase drastically. For example, for the case where backbone nodes (13% of all nodes) carry 79% of the total load S, the average load carried by them is about $6S/N$, while the nonessential nodes carry an average load of $0.2S/N$. Thus, when a nonessential node becomes a backbone node, the ratio β is of the order of 30, and due to heterogeneity of the nodes (each cluster is a scale-free network), the ratio can be as high as several hundred. On the other hand, the load flow in the backbone may be redistributed after the attack, leading to huge load changes as well. This accounts for the long tail in the distribution of the shifted ratio $\beta-1$. Although the ratio for a single node can be as high as several hundred, the number of such nodes can be several orders of magnitude smaller, as indicated by Fig. 4. We find, numerically, the effective value of $\beta\approx 2$ for a clustered network.

Thus, for a given value of λ, the optimal fraction of controlledly removed nodes is $f_c=1-\sqrt{\lambda/\beta}$. Noting that f can be written as $f=\eta M_r/M$, where η is the fraction of nonessential nodes, we have $M_{RC}=Mf_c/\eta$. Since M_{RC}, the optimal number of clusters where controlled removal occurs, assumes approximately the same value for different ways of selecting the clusters [Fig. 3(b)], our estimate for M_{RC} should practically hold for all three cases and it can thus be considered as a general theoretical prediction. For parameters used in Fig. 6(a), we have $\eta=0.87$. The predicted M_{RC} values are indicated by arrows in the figure for several λ values. They agree with the simulation results reasonably well.

We now summarize the steps of executing our strategy for preventing cascading breakdown in a complex clustered network. Assume that the network parameters λ, β, and η are available (either they are preassigned or they can be calculated when the network structure is known) and the backbone nodes in various clusters have been identified. The immediate response to an attack on some hub nodes in a particular cluster should be to calculate the distances between all other clusters to this cluster and assign indices I_M to these clusters. The critical cluster index $M_{RC}=\text{int}[M(1-\sqrt{\lambda/\beta})/\eta]$ is then calculated. Nonessential nodes in clusters whose indices satisfy $I_M \leqslant M_{RC}$ are removed. Cascading breakdown can then be avoided, where the resultant maximum value of G is given by $G_{\max}=1-f_c=\sqrt{\lambda/\beta}$. Numerical verification of our strategy is shown in Fig. 6(b), where the value of G versus λ is displayed. The result of executing our optimal strategy of controlled node removal is represented by the solid curve, while the dashed curve is predicted by the above physical analysis. We observe that, even when the node capacity parameter assumes the minimum value $\lambda=1$, our method can result in a connected component that contains more than 60% of the original nodes after an attack. In this sense, cascading breakdown has been effectively prevented. We emphasize that, given the structure of the network to be protected, the required computations in response to an attack can be done extremely efficiently, and the results of which can then be used for quick, controlled node removal so as to prevent possible cascading breakdown.

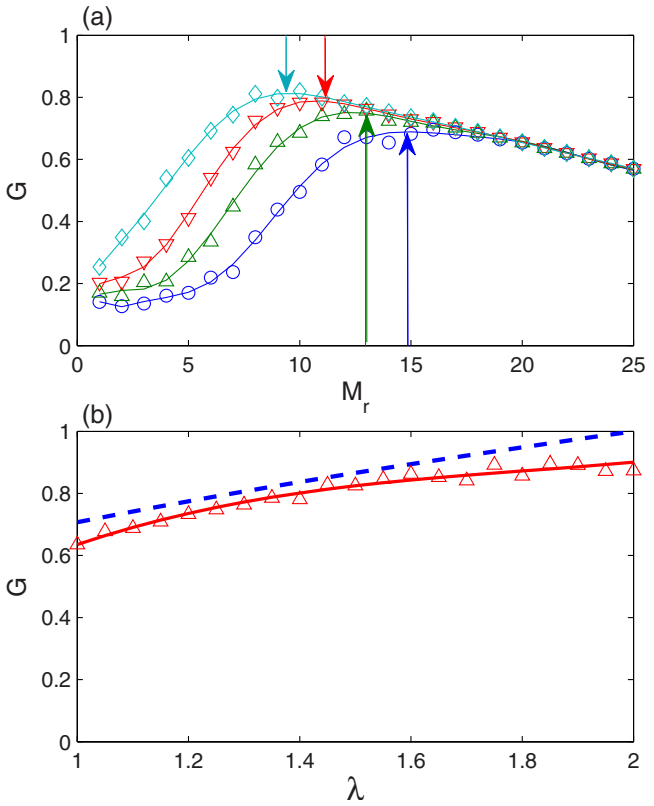

FIG. 6. (Color online) (a) G versus M_r for $\lambda=1.1$ (circles), 1.2 (upward triangles), 1.3 (downward triangles), and 1.4 (diamonds). The arrows indicate the predicted value of M_{RC}. (b) G versus λ for our strategy. The dashed line represents our theoretical prediction.

We thank K. Park for discussions. This work was supported by AFOSR under Grant No. FA9550-07-1-0045, by an ASU-UA Collaborative Program on Biomedical Research, by ONR through WVHTC (West Virginia High Technology Consortium Foundation), and by CityU of Hong Kong Grant No. SRG 7002274.

[1] P. Holme, Phys. Rev. E **66**, 036119 (2002).
[2] A. E. Motter and Y.-C. Lai, Phys. Rev. E **66**, 065102(R) (2002).
[3] Y. Moreno, J. B. Gómez, and A. F. Pacheco, Europhys. Lett. **58**, 630 (2002).
[4] X.-F. Wang and G.-R. Chen, IEEE Circuits Syst. Mag. **3**, 6 (2003); K.-I. Goh, D.-S. Lee, B. Kahng, and D. Kim, Phys. Rev. Lett. **91**, 148701 (2003); P. Crucitti, V. Latora, and M. Marchiori, Phys. Rev. E **69**, 045104(R) (2004).
[5] L. Zhao, K. Park, and Y.-C. Lai, Phys. Rev. E **70**, 035101(R) (2004).
[6] A. E. Motter, Phys. Rev. Lett. **93**, 098701 (2004).
[7] E. J. Lee, K.-I. Goh, B. Kahng, and D. Kim, Phys. Rev. E **71**, 056108 (2005); L. Zhao, K. Park, Y.-C. Lai, and N. Ye, *ibid.* **72**, 025104(R) (2005); D.-H. Kim, B. J. Kim, and H. Jeong, Phys. Rev. Lett. **94**, 025501 (2005); J. Xu and X.-F. Wang, Physica A **349**, 685 (2005); L. Huang, L. Yang, and K.-Q. Yang, Phys. Rev. E **73**, 036102 (2006); I. Simonsen, L. Buzna, K. Peters, S. Bornholdt, and D. Helbing, Phys. Rev. Lett. **100**, 218701 (2008).
[8] R. Albert, I. Albert, and G. L. Nakarado, Phys. Rev. E **69**, 025103(R) (2004); R. Kinney, P. Crucitti, R. Albert, and V. Latora, Eur. Phys. J. B **46**, 101 (2005).
[9] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[10] D. J. Watts, P. S. Dodds, and M. E. J. Newman, Science **296**, 1302 (2002); E. Ravasz, A. L. Somera, D. A. Mongru, Z. Oltvai, and A.-L. Barabási, *ibid.* **297**, 1551 (2002); V. Spirin and L. A. Mirny, Proc. Natl. Acad. Sci. U.S.A. **100**, 12123 (2003); A. E. Motter, T. Nishikawa, and Y.-C. Lai, Phys. Rev. E **68**, 036105 (2003); G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005); E. Oh, K. Rho, H. Hong, and B. Kahng, Phys. Rev. E **72**, 047101 (2005).
[11] W.-X. Qin and G.-R. Chen, Physica D **197**, 375 (2004); L. Huang, K. Park, Y.-C. Lai, L. Yang, and K. Yang, Phys. Rev. Lett. **97**, 164101 (2006); J.-J. Wu, Z.-Y. Gao, and H.-J. Sun, Phys. Rev. E **74**, 066111 (2006).