Final Accepted Version Chinese Journal of Aeronautics 28 (2015) xx-xx

Contents lists available at ScienceDirect

Chinese Journal of Aeronautics

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

Exploiting global information in complex network repair processes

WANG Tianyu[a,b], ZHANG Jun[a,b], Sebastian WANDELT[a,b,*]

[a] *School of Electronic and Information Engineering, Beihang University, 100191 Beijing, China*
[b] *Beijing Key Laboratory for Network-based Cooperative ATM, 100191 Beijing, China*

Abstract

Robustness of complex networks has been studied for decades, with a particular focus on network attack. Research on network repair, on the other hand, has been studied only very lately, given the even higher complexity and absence of an effective evaluation metric. A recently proposed network repair strategy is self-healing: This method aims to repair networks for larger components at a low cost only with local information.

In this paper, we discuss the effectiveness and efficiency of self-healing, which limits network repair to be a multi-objective optimization problem and makes it difficult to measure its optimality. This leads us to a new network repair evaluation metric. Since the time complexity of the computation is very high, we devise a greedy ranking strategy. Evaluations on both real-world and random networks show the effectiveness of our new metric and repair strategy. Our study contributes to optimal network repair algorithms and provides a gold standard for future studies on network repair.

Keywords: Complex network; Self healing; Optimality; Global information

1. Introduction

Many systems can be modeled as complex networks, where nodes represent elements of the system and links represent the relationship between elements. Examples include, but are not limited to, social[1, 2], economic[3, 4], traffic[5-7], biological[8, 9] and technological networks[10, 11]. Complex networks are often vulnerable under disruptions, e.g., caused by natural disasters or human intentional attacks[12, 13]. Therefore, research on network robustness and resilience has gained significant attention[14, 15]. Most researchers focus on network robustness, aiming to simulate network percolation processes; with the result that networks are often rather vulnerable[16-19]. One example of a high-impact network failure is the power grid break-up in Northeast of the U.S and parts of Canada on August 14th, 2003, which resulted from only a few power station failures. Another example is the air transportation delay occurred in Beijing on July 21st, 2012 caused by a rainstorm, which led to a widespread cascading failure of air traffic. Small events can leads to wide-ranging (cascading) failures.

Network repair, the inverse problem of network attack, is less studied, albeit the quick recovery of network functionality is a tremendous challenge. During a network disruption, either nodes or links can be attacked. Similarly, for network recovery, new links could be added between unaffected nodes or damaged nodes can be repaired[20, 21]. After a

*Corresponding author. Tel.: +86 10 82338036.
E-mail address: wandelt@buaa.edu.cn

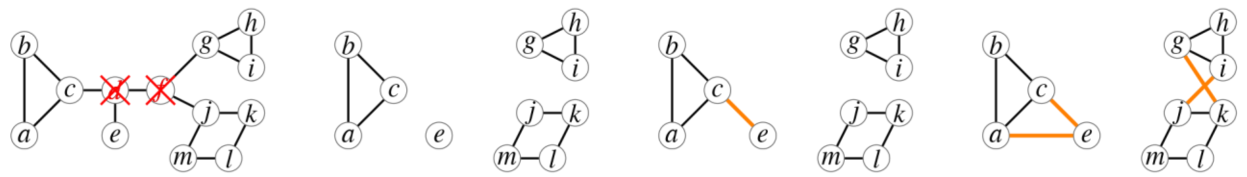
careful review of relevant research, we find that most researchers choose to attack nodes in a network and repair networks based on adding links. The latter, mostly because adding links between existing nodes should be much easier than building a new node. Under realistic condition, attacked nodes are often disabled forever, or even though they can recover from damage, the process is also very time-consuming, in which period the whole network would be influenced significantly. Therefore, the problem we study here is that under disruption, nodes in the networks would be attacked and that in the recovery process, we generate links between surviving nodes.

In this paper, we discuss a very recently proposed network repair strategy, self-healing[22]. This strategy repairs networks only using local information with the goal to obtain a larger size of the giant component (GC-size) and at the same time limits the cost of the repair. The cost here is defined to be the sum of the shortest path lengths of the added links in the original network and this definition is used in the evaluation section in our paper. However, this twofold goal of self-healing makes it a multi-objective optimization problem. Thus far, the optimality of self-healing has not been analyzed, i.e., the question, how much does the exploitation of local information affect the repair quality? Motivated by this point, we propose a shortest-path related network repair evaluation metric. The new evaluation metric is based on the shortest path length (SPL) of all node pairs in the network. In the new evaluation metric, the shortest path length between two nodes from different components is defined, since in traditional complex network theory, the shortest path length exists only between two connected nodes, no matter they are connected directly or indirectly. Besides, because of its constraint that only local information is needed, self-healing may not guarantee the repair quality. To assess the limitation of self-healing, global information is exploited in a new greedy ranking repair strategy. Experiments on real-world and random networks show that the new metric can evaluate the optimality of self-healing very well, and that the greedy ranking repair strategy restores networks' functionality in terms of the new metric.

The major contributions of this paper are summarized as follows:

1. We discuss the optimality of a recent network repair strategy, self-healing. Motivated by the limitation of self-healing, we create a new shortest-path related evaluation metric of network repair, which can measure the connectivity between all node pairs.
2. Based on the new evaluation metric, we propose a greedy ranking repair strategy, and provide an efficient implementation.
3. We evaluate the optimality of self-healing and the effectiveness of our strategy on six real-world networks and three kinds of random networks. Our results show that self-healing's optimality can be substantially improved in terms of the new metric.

This paper is structured as follows. In Section 2, we introduce the recently proposed network repair strategy self-healing. The limitation of this repair strategy is also discussed in this section. In Section 3, evaluation metrics of network repair are presented. A new evaluation metric and a greedy ranking strategy are proposed, and we make an improvement on the general strategy. In Section 4, both the optimality of self-healing and the effectiveness of our strategy are evaluated on six different real-world networks and three kinds of random networks. The conclusion and future work of this paper are presented in Section 5.



(a) Attack on an example network

(b) Network after the attack

(c) Best repaired network with self-healing repair method with $p = 0.5$ and $r_{max} = 2$

(d) A case of self-healing's results with $p = 0.8$ and $r_{max} = 3$

Figure 1: An attack on an example network (a), network after the attack (b), the best repair with self-healing while $p = 0.5$, $r_{\max} = 2$ (c) and a case of self-healing's results while $p = 0.8$, $r_{\max} = 3$ (d). Attacked nodes are crossed out. The orange link in (c) is the added link selected by self-healing repair strategy in the best case at a probability of 10%. When $p = 0.8$, $r_{\max} = 3$, there are nearly 200000 possible different repair results, and the orange links in (d) are added links with one of self-healing's results.

2. Self healing

In this section, both the details and limitation of self-healing will be discussed. Introduction of self-healing is presented in Section 2.1. The limitation is classified into three aspects: non-deterministic network repair (Section 2.2); constraints of local information and shortest path length (Section 2.3); redundancy of ineffective links (Section 2.4)

2.1. Introduction of self-healing

Self-healing is a very recently proposed strategy to repair complex networks. This strategy detects the level of the damage only relying on local information accessible to each node (its degree), and makes new connections with low cost. Therefore, in self-healing, each surviving node decides by itself whether it needs to generate a new link with another node. The process of self-healing can be divided into three parts:

1. An original network, the number of nodes to be attacked and two threshold q_c, r_{max} should be given. For each surviving node, the original degree k_{orig} and the degree after attack k_{dam} of each node should be recorded. q_c is the threshold for the fraction of remaining neighbors, and each node observes whether the fraction of its neighbors is less than this threshold. r_{max} means the limitation of the new links' distance.
2. For each surviving node, the value $q = k_{dam} / k_{orig}$ should be recorded and if $q < q_c$, the node is supposed to select a surviving node randomly to generate a new link.
3. For each new link, only if its distance (the shortest path length in the original network) is no longer than r_{max}, the link would be finally generated.

In Figure 1, the process of self-healing on an example network is shown. Figure 1 (a) shows an attack of two nodes on an example network; Figure 1 (b) is the network after the attack; the orange link in Figure 1 (c) is the added link with self-healing under the best condition at a probability of 10%, while there would be no link to be added at a probability of 90% (the parameters of self-healing here are $q_c = 0.5$ and $r_{max} = 2$); Figure 1 (d) presents one of the possible results with self-healing, while $p = 0.8$ and $r_{max} = 3$. The setting of parameters in Figure 1 (c) is the same as the evaluations in the self-healing paper, and is also used in the rest evaluations of this section.

An obvious advantage of self-healing is that it can repair networks very fast because only local information is needed. However, at a cost of the good response speed, the quantity of self-healing could not be guaranteed in terms of its evaluation metrics.

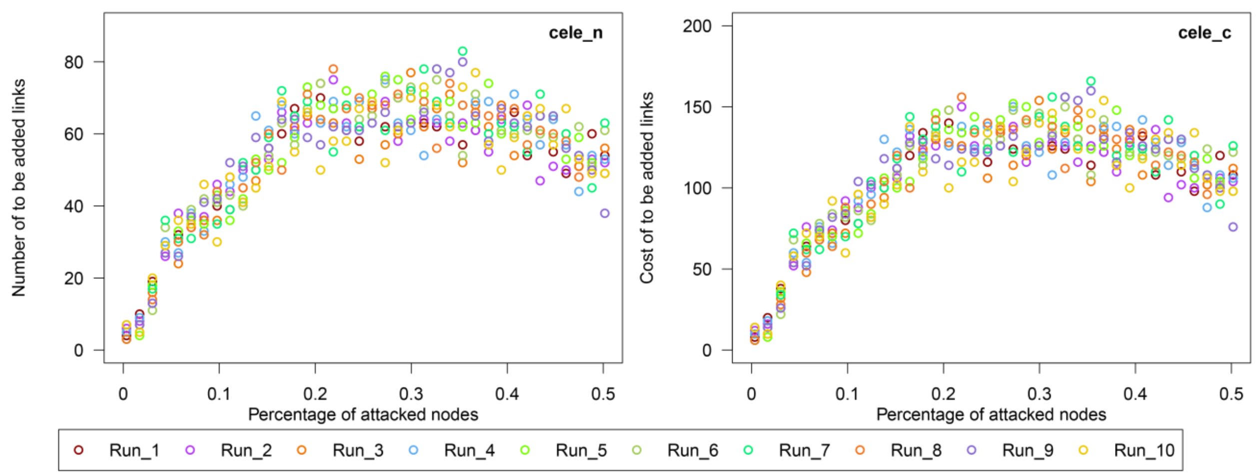


Figure 2: The results of self-healing repair strategy evaluated for ten times on a real-world network (celegansneural, introduced in Section 4.1). The left figure shows the number of added links as the percentage of attacked nodes changes; the right figure shows the cost of the added links as the percentage of attacked nodes changes. Different colors represent different evaluations of self-healing on the same network.

2.2. Non-deterministic network repair

Because of the limitation of self-healing, when one node decides to generate a new link, it can only select a surviving node randomly from the network. This selection criterion leads to very non-deterministic results, which make it confusing to identify whether a result is effective or not. To prove the randomness of self-healing, self-healing is evaluated on the same network (celegansneural, introduced in Section 4.1) for ten times to see how unstable the number of links and cost of the repair are, when the attack range changes. Here the attack range means the percentage of the attacked nodes' number to the total original network nodes' number, and the rest of this paper uses attack range with the same meaning.

Self-healing repairs the same network (celegansneural) for ten times when the attack range varies from 0 to 50%, and the results are shown in Figure 2. The results are considerably non-deterministic, and the extreme range of the results is more than 25%. As we can see from the figures, both in terms of the number of links and the repair cost, the range of the evaluation results increases with larger attack range. Since different evaluations of self-healing may get very different repair decisions, it is not convincing for practical application.

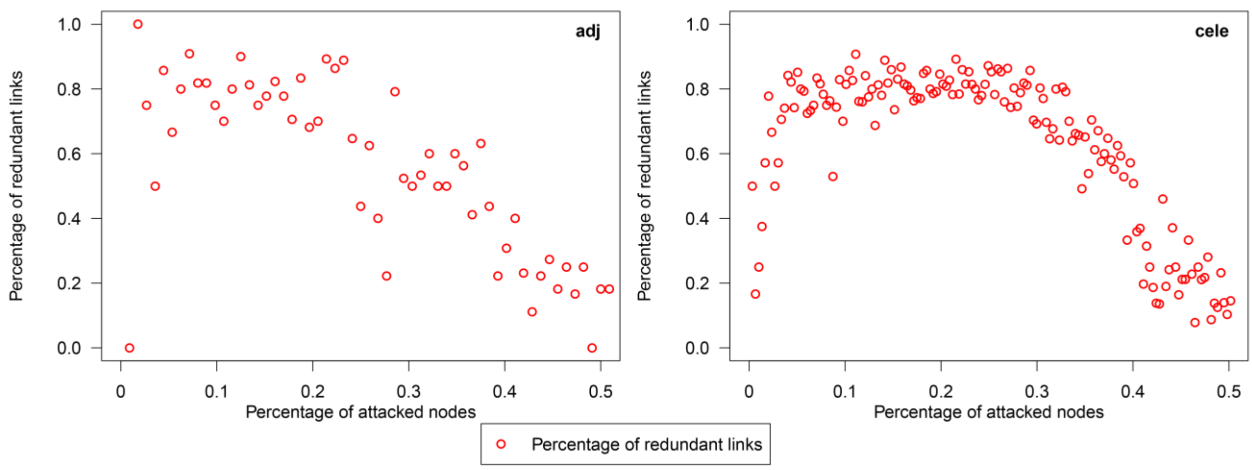


Figure 3: Percentage of redundant links as the percentage of attacked nodes of self-healing repair strategy changes on two real-world networks (adjnoun and celegansneural, introduced in Section 4.1).

2.3. Constraints of local information and short path length

Though only relying on local information is the superiority of self-healing, the absence of global information should influence the effectiveness of the repair significantly. As we can see in Figure 1, even in this very small network, self-healing could not guarantee its effectiveness. Without global information, each node may not recognize which nodes are more important to be linked. Therefore, only local information is not enough for a satisfying repair quality, and global information is necessary.

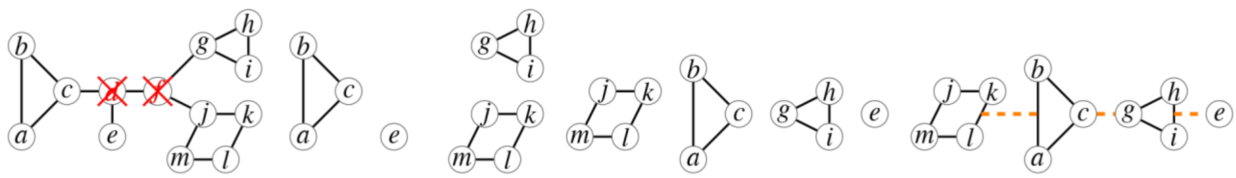
2.4. Redundancy of ineffective links

Since self-healing aims to repair a network for a larger *GC-size* at a low cost, it can be seen as a redundant link if the nodes of this link are already in the same component. This kind of links cannot increase the *GC-size* of the network. In other words, the resource of this link could be used in a more effective way. To show this deficiency of self-healing, it is used to repair two real-world networks (adjnoun and celegansneural, introduced in Section 4.1) to see the percentage of redundant links.

Figure 3 is the evaluation results of self-healing on two different networks, one containing 112 nodes and another with 297 nodes (adjnoun and celegansneural). The charts both show the percentage of redundant links when the attack range varies from 0 to 50%. As the charts report, self-healing in both networks generates a high percent of redundant links. In the first network with 112 nodes, the percentage of redundant links can be up to 100% of the total added links and only two points are below 10%. From 0 to 25% of the attack, nearly all the percentage of redundant links is more than 70%. Regarding another network with 297 nodes, nearly 80% of the first 50% attacks have more than 60% redundant links. Though the redundancy may strengthen the connectivity of some connected components, this is not the goal of self-healing, while its goal should be larger *GC-size* and low cost.

3. Greedy ranking repair strategy

Considering the limitation of self-healing discussed in Section 2, we attribute the unsatisfying results to the evaluation metrics used and the absence of global information. To simplify the problem, we want to modify the multi-objective optimization problem into a single-objective one. Therefore, in this section, we first concentrate on a new evaluation metric for network repair, and during the repair process each node should get global information. Based on the new evaluation metric, a greedy ranking repair strategy and an improvement on the strategy are proposed.



(a) Attack on the network (b) Network after the attack (c) Arrangement according to size (d) Strategy based on GC-size

Figure 4: An attack on the example network (a), network after the attack (b), (c) shows the arrangement of components according to their sizes and (d) is the optimal strategy based on GC-size. Attacked nodes are crossed out.

3.1. Consideration on evaluation metrics

First and naturally, one part, *GC-size*, is selected from the evaluation metrics of self-healing. When the evaluation metric is only *GC-size*, there is the optimal repair strategy, and the number of added links should be the number of components minus one. Links only need to be generated between components with the largest two *GC-sizes* after each repair of one link. For instance, Figure 4 shows the optimal repair process based on GC-size. In Figure 4 (c) a series of components are sorted according to their sizes. If the network is supposed to be repaired for larger GC-size, links only need to be generated between each component, and after adding only three links such as the orange dashed shown in Figure 4 (d), the network would be fully connected. Even though the network is fully connected, the functionality of the network may not recover from damage. For example, in terms of the shortest path length of some node pairs which can represent the efficiency of the network, the shortest path length of node m and node e could at most be 7 with optimal GC-size repair strategy, while the network only has 11 nodes totally. Besides, there could be varieties of repair strategies to make the network fully connected, but the functionality of the network would be quite different when different links are generated, which also makes it confusing to determine the optimal repair strategy.

Secondly, we consider to using the average shortest path length to be the evaluation metric. However, as the traditional definition in complex network theory, the shortest path length is only defined between two connected nodes. After some attempts on evaluation metrics, we come up with the total connectivity of the network to be the evaluation metric, which includes the influence of all the connections of node pairs in the network, no matter they are connected or disconnected. The connection of two connected is defined to be the shortest path length, while the connection between two disconnected nodes to be a constant. Considering that the definition should correspond to the reality, this kind of connection should not be shorter than or equal to the diameter of the original network, because if so, transportation between these two nodes would be achieved by other ways instead of this network before attack. Thus, we set the connection of two nodes from different components to be:

$$Connection = Diameter(G_{orig}) + 1 \qquad (1)$$

where $Diameter(G_{orig})$ means the diameter of the original network. The new metric is:

$$TotalConnectivity = \sum_{n_1 \in nodes(G), n_2 \in nodes(G)} Connection(n_1, n_2) \qquad (2)$$

where $Connection(n_1, n_2)$ means the connection between n_1 and n_2. Though the formulation of the new metric is all related to the connection in the network, the new metric is indeed influenced by the *GC-size* of the network. Because the connection between two unconnected nodes is longer than the diameter of original network, which should be much larger than two nodes' *SPL* from one component of an attacked network, if there are two relatively large components in the network, the optimal repair link would be one between these two components to change many connections between components into connections within the same component. Since the new evaluation metric can also stand for the efficiency of a network, it conforms to realistic requirements.

In addition, we want to concern on the average shortest path length (*ASPL*) as well as the *GC-size*. Since the shortest path length is defined in a fully connected network, we can only use the average shortest path length of the giant component to represent that of the whole network. Because of the different measurement units, the values of both *GC-size* and average shortest path length should be normalized. Therefore, we make a weighted sum of these two normalized values to be the evaluation metric as follows:

$$WeightedSum = w_1 * GCsize + w_2 * (1 - ASPL) \qquad (3)$$

where w_1 and w_2 are the weights of *GC-size* and *ASPL* respectively. It is easy to normalize *GC-size* since the value is already between 0 and 1, while the process of the normalization for *ASPL* is quite difficult. The maximum of *ASPL* could not be determined easily. If the maximum is changed according to different topological structures, it is obviously unfair for different kinds of repair strategies, because for a network with a fix number of nodes, the maximum should be the same. Finally, we set the value to be $(n+1)/3$, where *n* is the number of nodes in the network. This value is calculated according to the worst situation in terms of *ASPL* for a network of *n* nodes, which is a line network, and the *ASPL* of a line network with *n* nodes is $(n+1)/3$. However, after some evaluations of this metric, we found that the *ASPL* influences so much on the weighted sum that the optimal repair strategy of this evaluation metric nearly repairs networks only for the smallest *ASPL* of the giant component, which would ignore *GC-size*.

After the considerations on the evaluation metrics, we select the total connectivity to assess the quality of network repair.

3.2. Greedy ranking repair strategy based on total connectivity

After confirming our new evaluation metric, we consider to propose a new network repair strategy, which would behave well in terms of the new metric. The problem to be solved then is that when we want to generate a fixed number of links to an attacked network, which links should be generated to make the total connectivity of the network minimum. However, obtaining the optimal solution to this problem is NP-hard. For a network containing *n* nodes, if we want to generate *m* links to make the total connectivity to be minimum, the computation complexity to get the optimal result is $O\left(n^{2m+2}\right)$. This value can be derived as follows:

For a network containing n nodes, there are totally $\binom{n}{2}$ node pairs. Assuming the number of links in the network is l, there are $\left(\binom{n}{2} - l\right)$ links (the order of magnitude is n^2), which can be generated for repair. If m links are expected, there would be $\binom{\left(\binom{n}{2} - l\right)}{m}$ different combinations to be generated (the order of magnitude is n^{2m}). For each combination, the total connectivity needs to be computed to check its effectiveness. Computing the total connectivity needs to consider all node pairs in the network, so the order of magnitude to compute the total connectivity for one time is n^2. Therefore, the computation complexity to get the optimal result is $O\left(n^{2m+2}\right)$.

We have tried to find the optimal results for a very small network (karate, introduced in Section 4.1) containing 34 nodes at the length of 4 optimal links, whose running time is already nearly 6 hours.

Algorithm 1 Greedy ranking algorithm

Input: Original network G_{ori}; attacked network G_{att}; v, number of links to be added
Output: list of links to be added

for all $(n_1, n_2) \in N \times N \setminus edges(G) \setminus \{(n, n) \mid n \in N\}$ **do**
  Put (n_1, n_2) into $CandidateLinks$
end for
while $|AddedLinks| \neq v$ **do**
  $MinConnection = Diameter(G_{ori}) * |nodes(G_{att})|^2$
  for all $(n_1, n_2) \in CandidateLinks$ **do**
    $TotalConnectivity = 0$
    Let $Components(G_{att} \cup (n_1, n_2))$ be the list of $G_{att} \cup (n_1, n_2)$'s components.
    for all $i \in \{1, 2, ..., |Components(G_{att})|\}$ **do**
      if $|nodes(Subgraph(G_{att} \cup (n_1, n_2), Components[i]))| == 1$ **then**
        $TotalConnectivity += 0$
      else
        $TotalConnectivity = TotalConnectivity + averrage\_shortest\_path\_length(Subgraph(G_{att} \cup (n_1, n_2), Components[i])) * |Components[i]| * (|Components[i]| - 1)/2$
      end if
    end for
    for all $i \in \{1, 2, ..., |Components| - 1\}$ **do**
      for all $j \in \{1, 2, ..., |Components| - i - 1\}$ **do**
        $TotalConnectivity = TotalConnectivity + (Diameter(G_{ori}) + 1) * |Components[i]| * |Components[j + i + 1]|$
      end for
    end for
    if $TotalConnectivity < MinConnection$ **then**
      $BestLink = (n_1, n_2)$
      $MinConnection = TotalConnectivity$
    end if
  end for
  Add (n_1, n_2) into $AddedLinks$
  $G_{att} = G_{att} \cup (n_1, n_2)$
end while
Return $AddedLinks$

Forced by the long running time, we propose a greedy ranking repair strategy to get a relatively optimal solution to the problem. In the greedy ranking repair strategy, we divide the problem of *m* optimal links into *m* separated selections of one optimal link. Therefore, when *m* links are supposed to be generated in a network, we first sort all the candidate links according to the metric values (total connectivity) of the network with the specific one candidate link. Then we add the optimal one link to the network, and refresh the candidate links. We redo the two preceding steps until we finally add *m* links to the network. The computation complexity of this greedy ranking repair strategy is $O\left(n^4\right)$. This value can be derived as follows:

For each one optimal link, $\left(\binom{n}{2} - l\right)$ candidate links need to be compared (the order of magnitude is n^2). For each one candidate link, the total connectivity of the network with the candidate link should be computed for one time, and the order of magnitude to compute the total connectivity for one time is n^2. The order of magnitude to compute *m* relatively optimal links is $m * n^2$. Since the repair cost could be limited, *m* is often much smaller than *n*. Therefore, the computation complexity of the greedy ranking repair strategy is $O\left(n^4\right)$.

In our repair strategy, we do not calculate all node pairs' connectivity because the software package used has an efficient method to get the average shortest path length (*ASPL*) of each component in the network. Thus, we calculate the total connectivity according to the formulation as follows:

$$Connectivity_1 = \sum_{i \in \{1,2,\cdots,|Components(G)|\}} (ASPL(i) * |Components(i)| * (|Components(i)| - 1) / 2)$$

$$Connectivity_2 = \sum_{i \in \{1,2,\cdots,|Components(G)|\} \& i<j} (Diameter(G_{orig}) + 1) * |Components(i)| * |Components(j)|$$

$$TotalConnectivity = Connectivity_1 + Connectivity_2 \qquad (4)$$

where $|Components(G)|$ means the number of components in network G, $ASPL(i)$ means the ASPL of the ith component, $|Components(G)|$ means the number of nodes in ith component and $Diameter(G_{orig})$ is the diameter of the original network, which is used for the connection between two unconnected nodes in the network. In order to make the process of the greedy ranking repair strategy more clear, the pseudo code is presented in Algorithm 1. In this paper, all shortest path lengths are computed with NetworkX 1.9.1, which is a Python library for studying graphs and networks

Algorithm 2 Speed-up greedy ranking algorithm (1)

//Get the total connectivity of the network– $getConnectivity(G, d)$.
Input: Network G; d, the diameter of the original network
Output: The total connectivity of the network
Let $Components(G)$ be the list of G's components.
$TotalConnectivity = 0$
for all $i \in \{1, 2, ..., |Components(G)|\}$ **do**
  if $|nodes(Subgraph(G, Components[i]))| \neq 1$ **then**
    $TotalConnectivity = TotalConnectivity + averrage\_shortest\_path\_length(G_{sub}) * |Components[i]| * (|Components[i]| - 1)/2$
  end if
end for
for all $i \in \{1, 2, ..., |Components| - 1\}$ **do**
  for all $j \in \{1, 2, ..., |Components| - i - 1\}$ **do**
    $TotalConnectivity = TotalConnectivity + (d + 1) * |Components[i]| * |Components[j + i + 1]|$
  end for
end for
Return $TotalConnectivity$

3.3. Speed-up greedy ranking repair strategy

Though the greedy ranking repair strategy is much faster than the real optimal repair strategy, the computation complexity is still need to be optimized. The most natural idea to simplify the strategy is to avoid some tries on obviously ineffective links. Because the greedy ranking repair strategy needs to consider all node pairs in the network, we try to avoid some links, which can be easily identified not in the optimal repair links. In this speed-up strategy, we check the candidate links in order, and record the best one link and the total connectivity with this temporary best one link. When we check a new link, we estimate the upper-bound effectiveness of this new link to the whole network, and if this new link is better than the temporary best one, we compute the total connectivity of the network with this new link to check whether this new link is really better than that one.

The pseudo code of the speed-up strategy is shown as follows. The process of the strategy is divided into three parts: compute total connectivity of a network, Algorithm 2; select the best one link to be added, Algorithm 3; select fixed number of best links, Algorithm 4. The improvement of speed-up strategy comes from the second part.

Limited by the formulation of the effectiveness' upper-bound estimation, the speed-up strategy can only avoid some redundant computation before the network is fully connected. Because all links need to be checked before its effectiveness is really evaluated, the running time would be a bit longer than that of greedy ranking repair strategy after the network is fully connected. More improvement on the strategy in terms of computation complexity is our future work.

Algorithm 3 Speed-up greedy ranking algorithm (2)

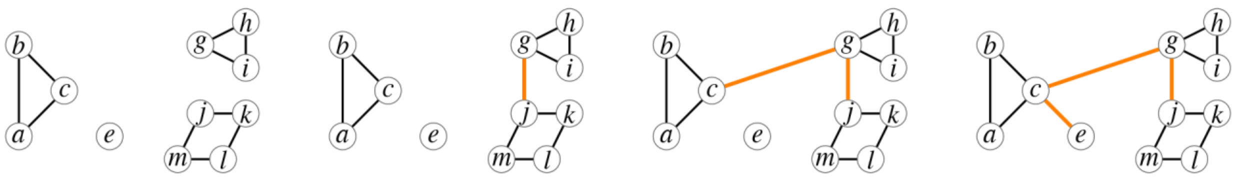
//Get the best one link– $GetBestLink(G, d)$.
Input: Network G; d, the diameter of the original network
Output: The best to be added link

Let L be a list of connected components in G, where each item is a set of nodes in the component
$N = nodes(G)$
$Best = (-1, -1)$
$BestDist = (d + 1) * |nodes(G)|^2$
if $|L| \neq 1$ **then**
  for all $(n_1, n_2) \in N \times N \setminus edges(G) \setminus \{(n, n) \mid n \in N)\}$ **do**
    if $node2Comp[n1] \neq node2Comp[n2]$ **then**
      $MaxImprovement = |node2Comp[n1]| * |node2Comp[n2]| * (d + 1)$
      if $(OrigDist - MaxImprovement) < BestDist$ **then**
        if $GetConnectivity(G \cup \{(n_1, n_2)\}) < BestDist$ **then**
          $BestDist = GetConnectivity(G \cup \{(n_1, n_2)\})$
          $Best = (n_1, n_2)$
        end if
      end if
    else if $n_1 \neq n_2$ **then**
      $MaxImprovement = |node2Comp[n1]| * |node2Comp[n1]| * (d + 1)$
      if $(OrigDist - MaxImprovement) < BestDist$ **then**
        if $GetConnectivity(G \cup \{(n_1, n_2)\}) < BestDist$ **then**
          $BestDist = GetConnectivity(G \cup \{(n_1, n_2)\})$
          $Best = (n_1, n_2)$
        end if
      end if
    end if
  end for
else
  for all $(n_1, n_2) \in N \times N \setminus edges(G) \setminus \{(n, n) \mid n \in N)\}$ **do**
    if $GetConnectivity(G \cup \{(n_1, n_2)\}) < BestDist$ **then**
      $BestDist = GetConnectivity(G \cup \{(n_1, n_2)\})$
      $Best = (n_1, n_2)$
    end if
  end for
end if
Return (n_1, n_2)

Algorithm 4 Speed-up greedy ranking algorithm (3)

//Get a number of links to be added–$GetToplinks(G_{ori}, G_{att}, v)$.
Input: Original network G_{ori}; attacked network G_{att}; v, number of links to be added
Output: list of links to be added

for all $i \in \{1, 2, ..., v\}$ **do**
  $BestLink = GetBestLink(G_{att}, Diameter(G_{ori}))$
  $G_{att}.add\_edges(BestLink)$
  Add $BestLink$ into $AddedLinks$
end for
Return $AddedLinks$



(a) Network after attack

(b) Network after repair with one link

(c) Network after repair with two links

(d) Network after repair with three links

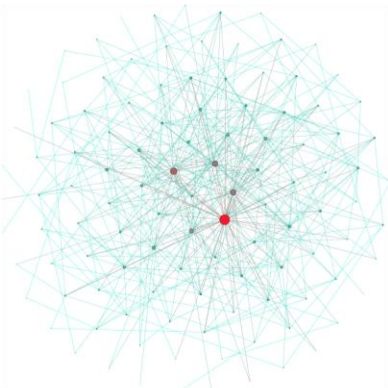
Figure 5: Example network after attack (a), the networks after repair with one to three links (b)–(d). Orange links are added links during repair process.

3.4. Effectiveness of greedy ranking repair strategy on a small example network

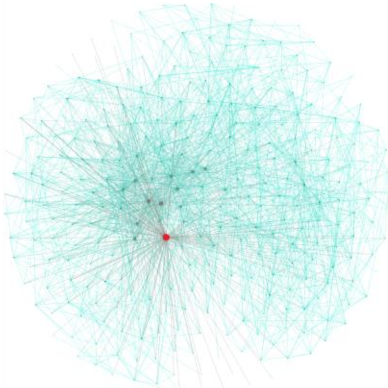
Because the speed-up strategy only avoids some links with poor effectiveness, the results of greedy ranking repair strategy and speed-up strategy are the same. Figure 5 shows the best three links with our repair strategy on the small attacked example network used in Section 2. The orange links are the added links. We can see that even though our evaluation metric is the total connectivity, the repair in this example network is also the optimal repair in terms of GC-size, which confirms that the new links between components are also important in terms of our metric

Table 1: Basic properties for the 6 networks used in our study. ASPL stands for average shortest path length. CC represents cluster coefficient.

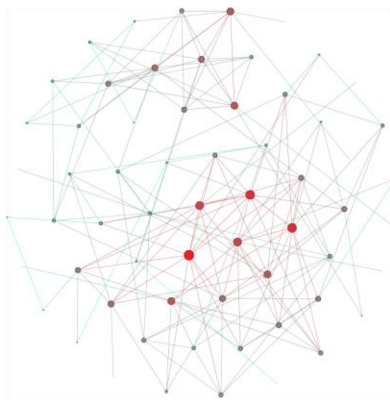
Network	|Nodes|	|Edges|	Avg. degree	Avg. betweenness	ASPL	CC	|Communities|
adjnoun	112	425	7.589	0.014	2.536	0.19	6
celegansneural	297	2148	14.465	0.005	2.455	0.308	4
dolphins	62	159	5.129	0.039	3.357	0.303	5
football	115	613	10.661	0.013	2.508	0.403	8
karate	34	78	4.588	0.045	2.408	0.588	4
lesmis	77	254	6.597	0.0219	2.641	0.736	6



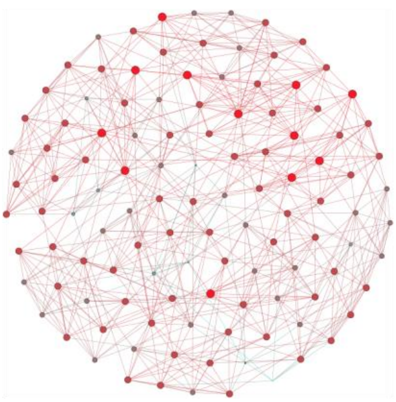
adjnoun: Common word adjacencies for the novel "David Copperfield"



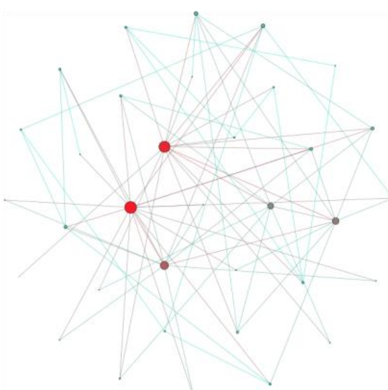
celegansneural: Neural network of C. Elegans



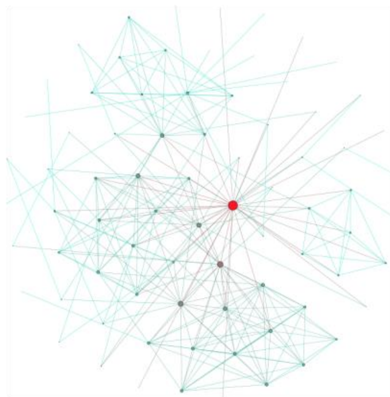
dolphins: Social network of frequent associations between 62 dolphins



football: Network of football games between Division IA colleges



karate: Friendships between members of a karate club at a US university



lesmis: Coappearances of characters in Victor Hugo's novel "Les Miserables"

Figure 6: Real-world networks used in the evaluation. The size of nodes is proportional to its degree in the network. The graph layout was created with the force-directed algorithm Fruchterman-Reingold.

4. Experimental evaluation

In this section, the optimality of self-healing and the effectiveness of the greedy ranking repair strategy are compared on six real-world networks and three kinds of random networks. The attack strategy during disruption is chosen according to degree, which means nodes with higher degree are attacked first. The relation between total connectivity and number of added links and that between total connectivity and cost of the repair are discussed in this section. In addition to the effectiveness of each strategy, the running time of each strategy is also reported at the end of each type of networks. All experiments are conducted on a laptop with Intel(R) Core(TM) i7-4610M CPU @ 3.00GHz processor and 4.00GB main memory. All implementations of network repair strategies were implemented in Python on Fedora 23.

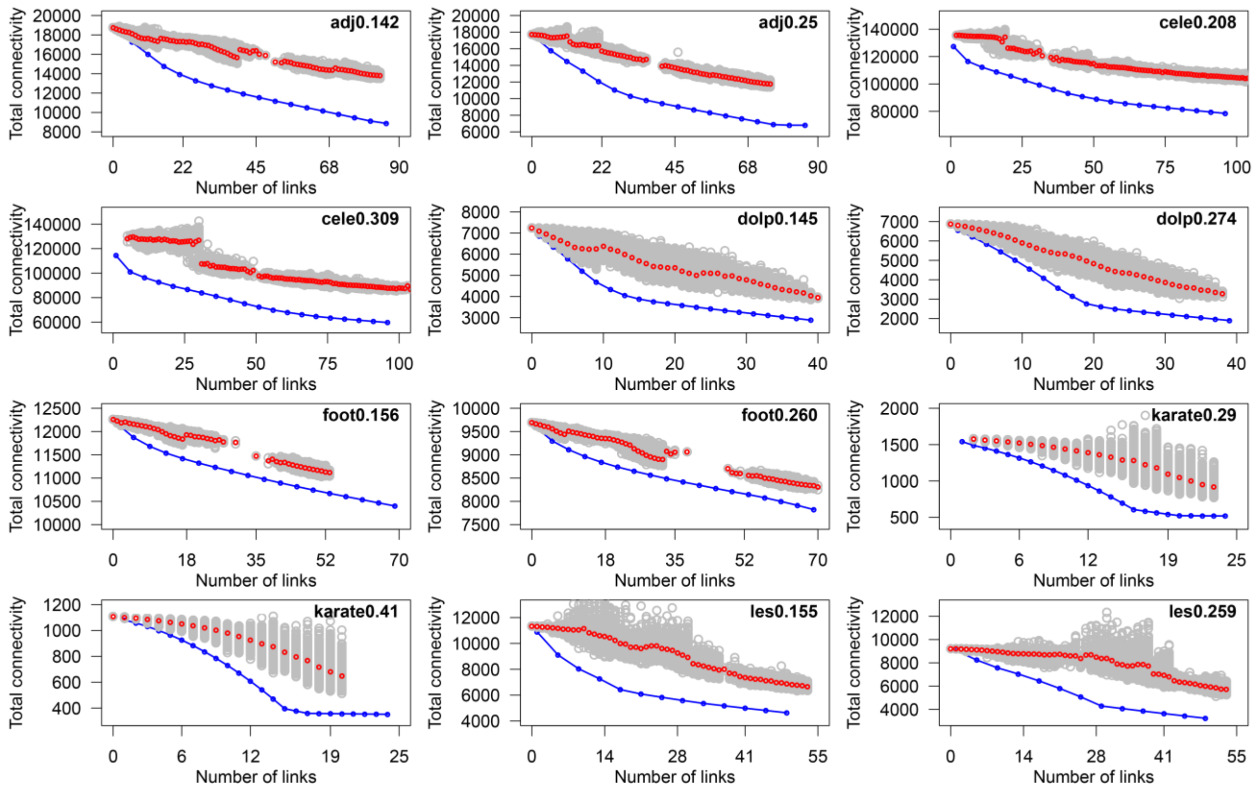


Figure 7: All figures show the total connectivity when the number of added links changes. Greedy ranking repair strategy and self-healing are evaluated on six real-world networks at two different attack ranges. Blue points represent the results of greedy ranking repair strategy. Self-healing is evaluated for 1000 times, and the gray points are the self-healing results. The red points are the averages of the results of self-healing. The upper bound of greedy strategy's link number is determined according to upper bound of self-healing's link number on the same network for 1000 times.

4.1. Evaluation on real-world networks

Table 1shows basic properties of the 6 real-world networks used in our study and Figure 6 are the visualizations of the 6 networks, where the size of a node is proportional to its degree. All networks are available for downloading at *https://networkdata.ics.uci.edu/index.php*. We can see from the table and figures that the structures of these 6 networks are quite diverse

To evaluate the optimality of self-healing and the effectiveness of greedy ranking strategy, we attack two different numbers of nodes on each of the 6 real-world networks and then repair them with both strategies. For self-healing, we change the parameters of self-healing and record all the results for 1000 executions, given that the outcome of self healing is non-deterministic. We vary parameter q_c from 0.1 to 0.9, and at the same time, vary r_{max} from 2 to 5. For the greedy ranking repair strategy, we gradually increase the number of to be added links. The upper bound of greedy strategy's link number is fixed to the maximum of self-healing's link number on the same network. Figure 7 reports the total connectivity as the number of added links changes. As we can see from the figure, in terms of number of added links and the total connectivity, the greedy ranking repair strategy obtains much better results than self-healing. For all 12 scenarios of our 6 networks, greedy ranking repair strategy's total connectivity is less than self-healing. Nearly for all the networks, greedy ranking repair strategy's total connectivity decreases sharply in the first stage, and when the number of added links become similar to the high-parameter ($q_c > 0.8, r_{\max} > 4$) self-healing results' link numbers, the slope of the total connectivity decreases. The large ranges of self-healing results also underline its strong non-deterministic behavior. It should be noted that in football network, the results of self-healing and greedy ranking repair strategy are very close, because the dense of football network is pretty high and even optimal results cannot decrease the total connectivity a lot.

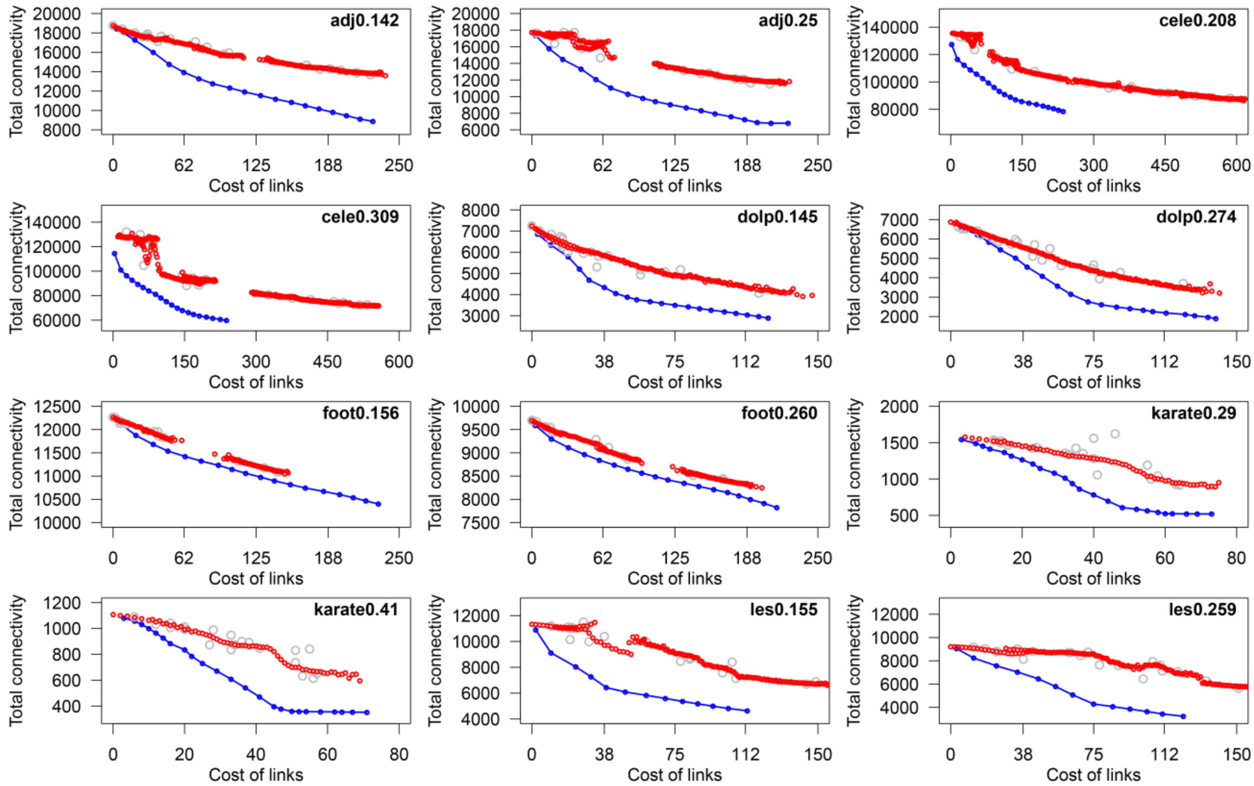


Figure 8: All figures show the total connectivity when the cost of added links changes. Greedy ranking repair strategy and self-healing are evaluated on six real-world networks at two different attack ranges. Blue points represent the results of greedy ranking repair strategy. Self-healing is evaluated for 1000 times, and the gray points are the self-healing results. The red points are the averages of the results of self-healing. The upper bound of greedy strategy's link number is determined according to upper bound of self-healing's link number on the same network for 1000 times.

Table 2: All running time for one link with greedy ranking repair strategy and running time of self-healing

Network–attack range	adj–0.142	adj–0.25	cele–0.208	cele–0.309	dolp–0.145	dolp0.274
greedy ranking (s)	81	42	1460	841	4.1	2.2
self-healing (s)	0.043	0.035	0.4	0.25	0.012	0.011
Network–attack range	foot–0.156	foot–0.260	karate–0.29	karate–0.41	les–0.155	les–0.259
greedy ranking (s)	90	50	0.17	0.08	9	5
self-healing (s)	0.053	0.045	0.004	0.004	0.02	0.015

Figure 8 shows the relation between the cost and total connectivity of both self-healing and greedy ranking repair strategy's results. As we can see, though the goal of our network repair strategy is not related to the cost of the repair, greedy ranking repair strategy's results are also better than self-healing's results in terms of cost, because more information is used. In terms of repair cost, self-healing' results are much more stable than those in terms of link numbers.

When it comes to running time, because the running time of greedy ranking repair strategy is almost linear as the increase of link numbers, the running time of only one added link with greedy ranking repair strategy is shown. Table 2 reports all the running time of one link in all networks with greedy ranking repair strategy and running time of self-healing. Self-healing has a big advantage in terms of running time compared with greedy ranking repair strategy.

4.2. Evaluation on random networks: ER, BA, WS

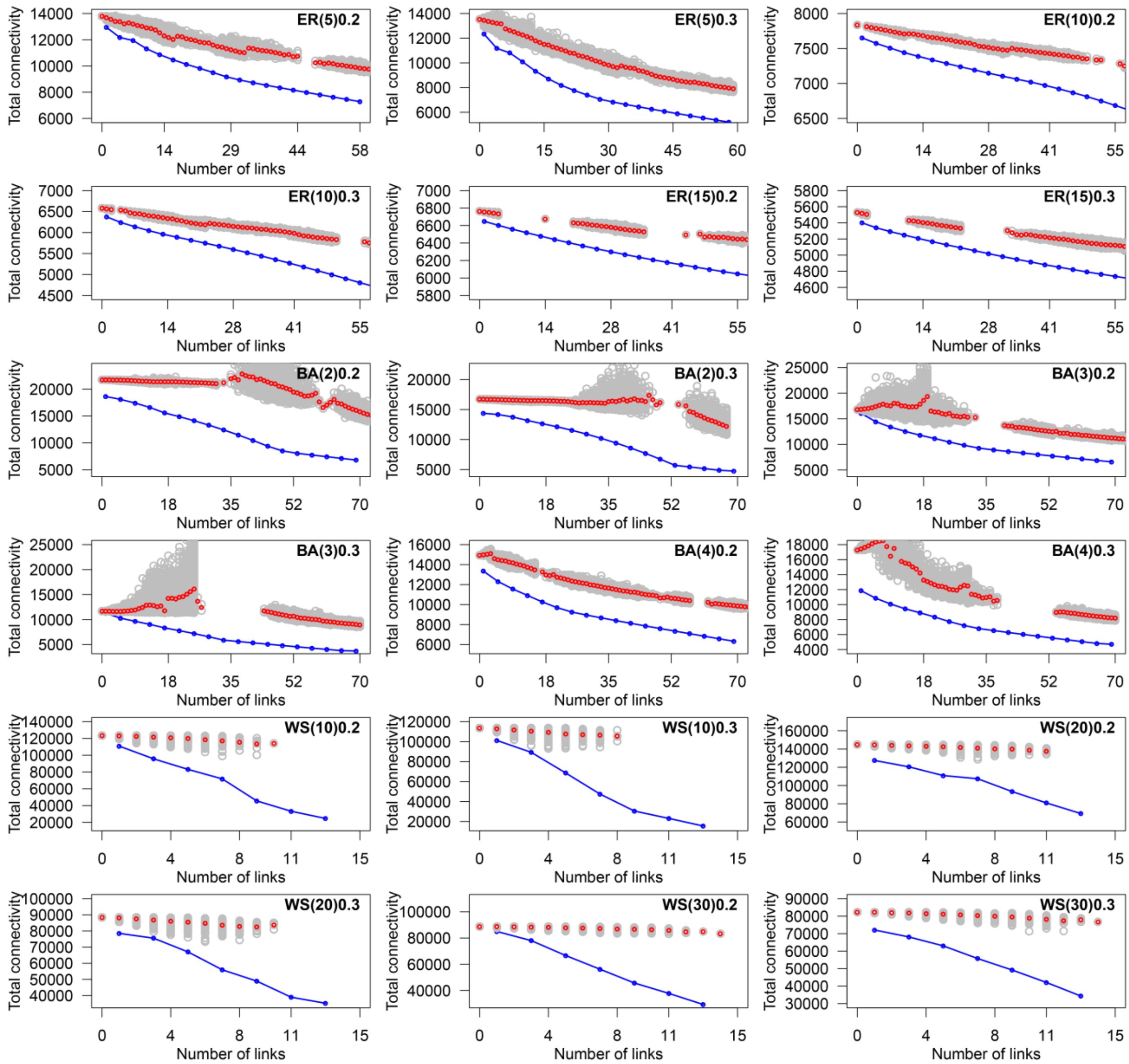


Figure 9: All figures show the total connectivity when the number of added links changes. Greedy ranking repair strategy and self-healing are evaluated on six real-world networks at two different attack ranges. Blue points represent the results of greedy ranking repair strategy. Self-healing is evaluated for 1000 times, and the gray points are the self-healing results. The red points are the averages of the results of self-healing. The upper bound of greedy strategy's link number is determined according to upper bound of self-healing's link number on the same network for 1000 executions. For the WS networks, because most connections between surviving nodes are longer than $r_{\max}$, self-healing could not generate many links even with relatively high probabilities.

In this part, the repair strategies are evaluated on three types of random networks (ER, BA and WS). For each type of random network, we set node number to be 100 and for each network the attack ranges are 20% and 30%. The introduction of these three types of random networks is following:

1. Erdos-Renyi network (ER). In our experiment, we choose the $G(n, p)$ model. In this model, the network consists of *n* nodes, which is connected randomly. Each edge is linked in the network with probability *p* independent from every other edge. And all networks with *n* nodes and *m* edges here have equal probability:

$$p(n,m) = p^m (1-p)^{\binom{n}{2}-m} \tag{5}$$

In our evaluation, we set *p* to be 5%, 10% and 15%.

2. Barabasi-Albert network (BA). BA network, whose degree distribution follows a power law, is absolutely different from ER network. In this network model, there are also 2 parameters in it: n (number of nodes) and m (number of edges generated by one new node). The new added node has a linear preferental

$$p(k_i) = \frac{k_i}{\sum_j k_j} \tag{6}$$

where k_i means degree of node i. In our evaluation, the parameter m is 2, 3 and 4 respectively.

3. Watts-Strogatz network (WS). In WS network model, there are 3 parameters in it: n (number of nodes), m (initial number of each node's neighbors) and p (probability of each node rewiring). We set m to be 2 and p to be 10%, 20% and 30% in our evaluation.

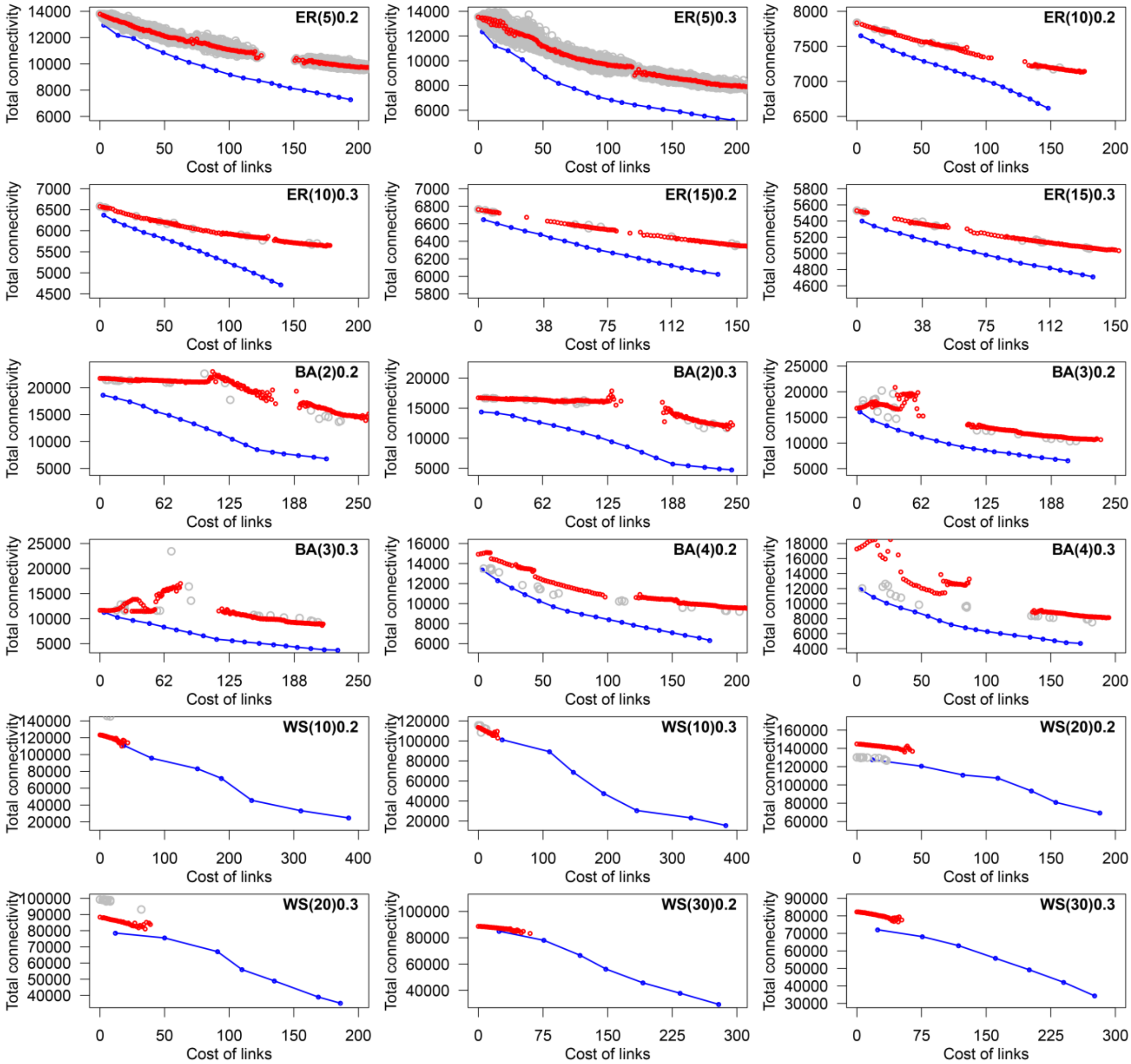


Figure 10: All figures show the total connectivity when the cost of added links changes. Greedy ranking repair strategy and self-healing are evaluated on six real-world networks at two different attack ranges. Blue points represent the results of greedy ranking repair strategy. Self-healing is evaluated for 1000 times, and the gray points are the self-healing results. The red points are the averages of the results of self-healing. The upper bound of greedy strategy's link number is determined according to upper bound of self-healing's link number on the same network for 1000 executions. For the WS networks, because most connections between surviving nodes are longer than $r_{\max}$, self-healing could not generate many links even with relatively high probabilities.

Table 3: All running time for one link with greedy ranking repair strategy and running time of self-healing.

Network–parameter –attack range	ER–5–0.2	ER–5–0.3	ER–10–0.2	ER–10–0.3	ER–15–0.2	ER–15–0.3
greedy ranking (s)	29	16	33	19	70	34
self-healing (s)	0.05	0.04	0.06	0.05	0.065	0.06
Network–parameter –attack range	BA–2–0.2	BA–2–0.3	BA–3–0.2	BA–3–0.3	BA–4–0.2	BA–4–0.3
greedy ranking (s)	55	25	55	26	77	28
self-healing (s)	0.035	0.030	0.042	0.032	0.053	0.033
Network–parameter –attack range	WS–10–0.2	WS–10–0.3	WS–20–0.2	WS–20–0.3	WS–30–0.2	WS–30–0.3
greedy ranking (s)	68	35	45	33	47	32
self-healing (s)	0.018	0.014	0.024	0.016	0.028	0.018

Similar to the evaluation on real-world networks, Figure 9 shows the total connectivity as link number differs. Under all conditions of random networks, greedy ranking repair strategy's results are better than self-healing. We can see that for ER networks, when the parameter setting is 10 or 15, which means each edge is linked with probability of 10% or 15% in the network, the trends of the two strategies' results are similar to that of football network, because these two parameter settings make the networks' dense so high that even the optimal repair could not decrease the total connectivity a lot. For WS networks, because the attacked networks have many components and the connection between components is pretty long (nearly all the candidate links' distance is longer than the threshold $r_{\max}$), self-healing can hardly decrease the total connectivity. This point also highlights that only local information cannot repair networks effectively.

Figure 10 reports the total connectivity as the cost changes. Greedy ranking repair strategy behaves better even in terms of cost, which is similar to the results of real-world networks. As shown in the WS networks, because of the constraints of self-healing, even with large parameters, it cannot decrease the total connectivity a lot.

Running time of random networks is shown in the way similar to that of real-world networks. Since the running time of greedy ranking strategy in random networks is still linear to the number of links, the running time of only one link with our strategy for each network is shown in Table 3. Because the WS networks have more components, speed-up strategy is more effective in these networks in terms of running time.

4. Conclusions and Future Work

In this paper, we discussed the recently proposed network repair strategy self-healing and its optimality. Motivated by the limitation of this strategy, a new metric is devised to evaluate network repair process and, according to this metric, a greedy ranking repair strategy and an improvement on the strategy are designed. The optimality of self-healing and the effectiveness of our strategy were evaluated on six real-world networks and three types of random networks with different parameters. Our results show that self-healing can be significantly improved by exploiting global information. This leads towards an interesting sweep point for research between local and globally optimal repair. Table 4 reports the effectiveness of self healing for real-world networks. The value we choose to compare in this table is the area between the trend line of the connectivity value and x-axis, which is the accumulated effectiveness of the strategy. As the table shows, the results are recorded as original connectivity, greedy ranking connectivity and self-healing connectivity. Each value represents the area between the trend line of the connectivity points and the x-axis, while the connectivity points of self-healing are the average values of 1000-time executions. The ranges of all these three kinds of area on the x-axis are the maximums of the self-healing link numbers. Therefore, the unit of these values is "$number * connectivity$". The red digits represent improvement, defined as follows:

$$\begin{aligned}&\mathrm{Im}\,provement(\%) = (SelfHealingConnectivity - GreedyConnectivity)\\&/(OriginalConnectivity - GreedyConnectivity)\end{aligned} \tag{7}$$

The value of $\mathrm{Im}\,provement(\%)$ should be between 0% and 100%. When $\mathrm{Im}\,provement(\%)$ is close to 100%, it means that self-healing repairs the network ineffectively; when $\mathrm{Im}\,provement(\%)$ is close to 0%, it means that self-healing's result is nearly optimal. It is shown that all values of the improvement are more than 40% in the 12 networks; 6 of them are more than 50%; 2 of them are more than 60%. The results show that self-healing's optimality can be improved significantly, by possibly exploiting non-local information. In conclusion, this paper leads to an effective evaluation metric on network repair, and sheds light on optimality of network repair strategies.

Table 4: Accumulated effectiveness of greedy ranking repair strategy and self-healing is shown. The self-healing connectivity is obtained by the average values of 1000-time results. And the red digits means the improvement our strategy make beyond self-healing.

Network–attack range	adj–0.142	adj–0.25	cele–0.208	cele–0.309	dolp–0.145	dolp0.274
original connectivity (number*connectivity)	1555586	1329900	12757398	11785410	260640	233750
greedy ranking connectivity (number*connectivity)	1024614.5	780139	8807002.6	7036666	144097	120344
self-healing (number*connectivity)	1268889	1032292	11093522	9726875	199542	174737
Improvement percent (%)	46.0	45.9	57.9	56.7	47.6	48.0
Network–attack range	foot–0.156	foot–0.260	karate–0.29	karate–0.41	les–0.155	les–0.259
original connectivity (number*connectivity)	600789	591273	36593	22180	555807	451927
greedy ranking connectivity (number*connectivity)	551687	522872.5	21949	13787	307373.5	262650.5
self-healing (number*connectivity)	575275	552372	30743	18621	453597	398571
Improvement (%)	46.5	41.8	60.1	57.6	58.9	71.8

Air transport networks are often vulnerable against emergencies or extreme weather, which are impractical to escape from. Therefore, an effective and efficient repair strategy is significantly important for the transport networks to restore their function. This paper developed an evaluation metric to assess the functionality of the networks and presented a greedy ranking strategy to help networks heal from the damages.

Our work can be improved in the following aspects:

1. Our current greedy ranking repair strategy checks all the candidate links, but only one link would be selected once, so there is no need to try each candidate link. Though the speed-up ranking strategy can avoid some ineffective links, but the speed-up process is only effective before the network is fully connected. Therefore, we have two choices to avoid more links: a) make a more accurate estimation formulation, because current estimation formulation is very natural to come up with; b) develop another technique for recognizing ineffective candidate links after the network is fully connected.
2. Simplify the criterion when evaluating each candidate link. For current strategy, the criterion is to calculate the total connectivity after adding the candidate links. Since most of current running time is spent on this process, if we want to improve the running time of our strategy, the criterion must be simplified. Therefore, we will try to find a new criterion with similar effectiveness of the total connectivity, but with significantly reduced computation complexity.

Acknowledgements

This study is supported by the Research Fund from National Natural Science Foundation of China (Grant No. 61521091, No. 61650110516, and No. 61601013).

References

[1] Parsons S, Atkinson PM, Simperl E, Weal M, Thematically analysing social network content during disasters through the lens of the disaster management lifecycle, in: Proceedings of the 24th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2015, pp. 1221–6.
[2] Ofei-Dodoo S, Medvene LJ, Nilsen KM, Smith RA, Dilollo A, Exploring the potential of computers to enrich home and community-based services clients' social networks, Educ Gerontol 41 (3) (2015) 216–25.
[3] Padilha-Feltrin A, Quijano Rodezno DA, Mantovani JRS, Volt-var multiobjective optimization to peak-load relief and energy efficiency in distribution networks, IEEE T Power Deliver 2015; 30 (2): 618–26.
[4] Havlin S, Kenett DY, Cascading failures in interdependent economic networks, in: Proceedings of the International Conference on Social Modeling and Simulation, plus Econophysics Colloquium 2014, Springer, 2015, pp. 87–97.
[5] Guan X, Zhang X, Han D, Zhu Y, Lv J, Su J, A strategic flight conflict avoidance approach based on a memetic algorithm, Chinese J Aeronaut 2014; 27 (1): 93–101.

[6] Li S, Xu X, Vulnerability analysis for airport networks based on fuzzy soft sets: From the structural and functional perspective, Chinese J Aeronaut 2015; 20 (3): 780–8.
[7] Wandelt S, Sun XQ, Evolution of the international air transportation country network from 2002 to 2013., Transport Res E-Log 2015; 82: 55-78.
[8] Gallos LK, Makse HA, Scaling theory of transport in complex biological networks., P Natl Acad Sci USA 2007; 104 (19): 7746–51.
[9] Talikka M, Boue S, Schlage WK, Causal biological network database: A comprehensive platform of causal biological network models focused on the pulmonary and vascular systems, Computational Systems Toxicology 2015; 65–93.
[10] Gan C, Yang X, Liu W, Zhu Q, Jin J, He L, Propagation of computer virus both across the internet and external computers: A complex-network approach, Commun Nonlinear Sci 2014; 19 (8): 2785–92.
[11] Liu N, An H, Gao X, Li H, Hao X, Breaking news dissemination in the media via propagation behavior based on complex network theory, Physica A 2016; 453: 44–54.
[12] Dong G, Du R, Hao H, Tian L, Modified localized attack on complex network, EPL 2016; 113 (2): 28002.
[13] Nie T, Guo Z, Zhao K, Lu ZM, The dynamic correlation between degree and betweenness of complex network under attack, Physica A 2016; 457: 129–137.
[14] Sun SW, Ma YL, Li RQ, Wang L, Xia CY, Tabu search enhances network robustness under targeted attacks, Physica A 2016; 446: 82–91.
[15] Alenazi MJ, Sterbenz JP, Evaluation and comparison of several graph robustness metrics to improve network resilience, in: Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on, IEEE, 2015, pp. 7–13.
[16] Kinney R, Crucitti P, Albert R, Latora V, Modeling cascading failures in the north american power grid, Physics of Condensed Matter 2015; 46 (1): 101–7.
[17] Simonsen I, Buzna L, Peters K, Bornholdt S, Helbing D, Transient dynamics increasing network vulnerability to cascading failures., Phys Rev Lett 2008; 100 (21): 2539–41.
[18] Zhou J, Huang N, Sun X, Wang K, A new model of network cascading failures with dependent nodes, in: Reliability and Maintainability Symposium (RAMS), 2015 Annual, 2015, pp. 1–6.
[19] Wandelt S, Sun XQ, Cao XB, Computationally efficient attack design for robustness analysis of air transportation networks, Transportmetrica A 2015; 11 (10): 939-66
[20] Wang J, Qiao C, Yu H, On progressive network recovery after a major disruption, Proceedings - IEEE INFOCOM 2011; 8 (1): 1925–33.
[21] Liu X, Yuan Y, A novel dynamic method in distributed network attack-defense game, Math Probl Eng 2015 2015; 1–7.
[22] Gallos LK, Fefferman NH, Simple and efficient self-healing strategy for damaged complex networks, Phys Rev E 2015; 92: 052806.