Robust Network Construction against Intentional Attacks

Hui Wang[‡], Wanyun Cui[†], Yanghua Xiao[†] , Hanghang Tong[§]
[‡]Shanghai University of Electric Power, Shanghai, China
[†]School of Computer Science, Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China;
[§]Arizona State University, USA;
[‡]wanghuilele@gmail.com [†]wanyuncui1@gmail.com, shawyh@fudan.edu.cn [§]hanghang.tong@asu.edu

Abstract*—Building large robust networks against intentional attacks has been well recognized important in designing infrastructure and cyberphysical systems. In some applications, complex networks have to be constructed under some resource constraints, degree sequence constraints on vertices being a popular one. In this paper, we study a novel and challenging problem:** *constructing large robust networks with degree sequence constraints against intentional attacks. We propose a heuristic approach with several desirable properties. Using both synthetic and real networks we show that our method improves the robustness of large networks substantially.**

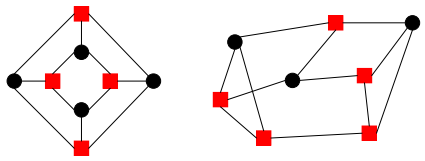
I. Introduction

Building large robust complex networks against intentional attacks has been well recognized important in designing infrastructure and cyberphysical systems, such as transportation networks [35], power grids [27], Internet [6], [34], water distribution networks [18] and wireless sensor network [26]. In an *intentional attack* [1], [15], an attacker selects and attacks some nodes in a network that play a vital role in maintaining the connectivity of the network, until either the network is scrapped to the extent imprecated by the attacker, or the attacker runs out of the attack budget.

Degree is one of the most apparent measures on the vitality of a node maintaining the connectivity of a large network. Intentional attacks guided by vertex degrees, also known as *highest-degree attacks* (*HDA* for short) [14], which target at vertices of the highest degree, have been widely used in the evil practice. Constructing a complex network against HDA thus has been an important task in designing a robust real complex networked systems.

The robust network construction is usually under the degree distribution constraint [14], [16]. First, *some real applications have strong constraints on degree distribution of the network*. For example, to build a transportation network on a set of airports, each airport has a limited capacity often in terms of the maximum number of take-offs and landings. The degree constraint is necessary to reflect the real traffic constraint. Second, *degree distribution is one of the most important characteristics of large networks*. Degree distribution is critical for many important problems on networks, such as information diffusion [11] and disease epidemic spreading [9], [5], [3].

Given a network, the degree distribution can be precisely represented as a degree sequence, that is, the sorted list of vertex degrees in descending order. Thus, *can we improve the robustness of a network while preserving the degree sequence?*



(a) G_1 (b) G_2

Fig. 1. Two 3-regular graphs with the same degree sequence and different robustness

The answer is positive, which is illustrated in Figure 1. In Figure 1, graphs G_1 and G_2 have the same size and both are 3-regular, that is, each vertex has degree 3. Hence, G_1, G_2 have the same degree sequence. However, G_2 is more resilient to vertex removal than G_1. To break all edges, at least 4 vertices, either all the square ones or all the circle ones, need to be removed from G_1. In G_2, at least 5 vertices (those square ones, for example) need to be removed. This simple example illustrates that networks with the same degree sequence and the same degree distribution may exhibit different resilience to vertex removal attacks. The robustness of a network depends on not only the degree distribution but also the linkage structure. It is possible to improve the robustness of a network and retain the degree distribution.

In this paper, we tackle the problem of *designing an efficient and effective approach to improve the robustness of a network against HDA while preserving the degree sequence*. We observe that, in many real networks, many vertices of small degrees and within small clusters heavily rely on some hub vertices with high degrees to connect to the rest of the network. When the hub vertices are attacked, the network can be easily broken into small pieces. Our idea is to adjust the network structure to reduce the influence of hub vertices on the connectivity of the rest of the network. At the same time, we need to accommodate the degree sequence constraint.

II. Problem Formulation

In this section, we first give the basic notations. Then, give the formal definition of the major problem solved in this paper. Finally, we present our algorithm framework to solve the problem.

TABLE I. NOTATIONS TABLE

Notations	Meaning
G	a simple graph
$\mathbf{d}$	a non-increasing degree sequence
$\mathbf{d}(G)$	degree sequence of graph G
$\mathcal{G}(\mathbf{d})$	the set of all graphs realizing $\mathbf{d}$
$deg(v)$, $deg_G(v)$, $deg_{\mathbf{d}}(v)$	degree of node v
$\tau(G)$	size of the minimal vertex cover of G
$\rho(G)$	cycle coverage of graph G

Algorithm 1 Overall Framework($\mathbf{d}$)

Input: $\mathbf{d}$: a graphical degree Sequence;
Output: A simple graph realizing $\mathbf{d}$ with optimized robustness;
{Step 1: Construct a connectivity skeleton G_c by $\mathbf{d}$;}
{Step 2: Residue degree sequence matching;}
1: $\mathbf{d}' \leftarrow \mathbf{d} - \mathbf{d}(G_c)$;
2: Sort $\mathbf{d}'$ into the decreasing order of degrees;
3: $G_r \leftarrow DegreeMatching(\mathbf{d}', G_C)$;
4: **return** $G_r \cup G_c$;

A. Notations

We give notations in Table I. For the sake of simplicity, in this paper, we discuss simple graphs only. However, our results can be easily extended to some other types of graph. We use the terms graph and network exchangeably. Let $G(V, E)$ be a simple graph, where $V = \{v_1, \ldots, v_N\}$ is a set of vertices and $E \subseteq V \times V$ is a set of edges. Let $N = |V|$. Denote the degree of a vertex by $deg(v_i)$. The *degree sequence* of G is the sorted list of the vertex degrees in ascending order, denoted by $\mathbf{d}$. We use $deg_G(v)$ to denote the degree of vertex v in graph G, and use $deg_{\mathbf{d}}(v)$ to denote the degree of vertex v in degree sequence $\mathbf{d}$. Without loss of generality, we assume that $v_1, v_2, \ldots, v_N$ are in degree non-increasing order, that is, $d(v_i) \geq d(v_{i+1})$ for $1 \leq i < N$. $\mathbf{d}(G) = (deg(v_1), \ldots, deg(v_N))$ is called the *degree sequence* of G. A non-increasing sequence $\mathbf{d}$ of positive integers is called a *graphical sequence* if there exists a simple graph G such that $\mathbf{d}(G) = \mathbf{d}$. We also say G *realizes* $\mathbf{d}$. Denote by $\mathcal{G}(\mathbf{d})$ the set of all graphs that realize $\mathbf{d}$. $\mathcal{G}(\mathbf{d}) = \emptyset$ if no graph realizes $\mathbf{d}$. For example, no simple graph can realize $\mathbf{d} = 3, 2, 1$.

B. Problem model

To define our problem, we first need to measure the robustness of a network. In this paper, we measure the robustness of a graph by the minimal number of vertices to be removed to destruct a network to a certain degree. Then, we formalize the major problem to be solved in the paper.

The robustness of a network against HDA can be measured by the minimum number of vertices having the highest degrees that have to be removed so that the graph is fragmented sufficiently. In practice, a real network often loses its functions when the largest connected component is smaller than ϵN, where $\epsilon \in (0, 1)$. We denote by $f_\epsilon(G)$ the minimum number of vertices that need to be removed so that the Maximum Connected Component (MCC) in the remaining graph is smaller than ϵN, where $\epsilon \in (0, 1)$ is a user-specified parameter. Now, we are ready to define our problem.

Definition 1 (Problem definition): Given a graph G, the problem of *improving G against HDA with degree sequence constraint* is to find a graph G' such that $\mathbf{d}(G) = \mathbf{d}(G')$ and $f_\epsilon(G')$ is maximized.

One straightforward method is to enumerate all members in $\mathcal{G}(\mathbf{d})$. Unfortunately, we have the following result.

Lemma 1 (Upper bound of $|\mathcal{G}(\mathbf{d})|$): For a graph G, $|\mathcal{G}(\mathbf{d})| \leq \binom{|V|(|V|-1)/2}{|E|}$. The upper bound is reachable.

Lemma 1 hints that the search space may be exponential to the size of the network. Thus, exhaustively enumerating all graphs in $\mathcal{G}(\mathbf{d})$ to find the one of the optimal robustness cannot be scalable on large networks. This observation motivates us to design an efficient heuristic method.

C. Solution Framework

We propose a two-step algorithm framework (shown in Algorithm 1) to address the major issues in our problem definition: *robustness optimization* and *preservation of degree sequence*.

In the first step, we construct a connectivity skeleton G_c that is accounting for the robustness of the network. In this step, we expect G_c is robust and for each vertex u, $deg_{G_C}(u) \leq deg_{\mathbf{d}}(v)$. In the process, for a vertex v, if the degree of v in the degree sequence is $deg(v)$, and the number of edges assigned to v so far is $deg(v')$ $(deg(v') \leq deg(v))$, the difference $deg(v) - deg(v')$ is called the *residue degree*, which is the degree to be realized in the second step. We will give two skeleton construction approaches in Section III.

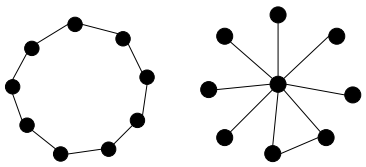
In the second step, we calculate the residue degree sequence $\mathbf{d}'$ by $d'_i = deg(v_i) - deg_{G_c}(v_i)$ for each $d'_i \in \mathbf{d}'$. Then, we try to construct a graph G_r to satisfy $\mathbf{d}'$ with the constraint that each edge in G_r is not in G_c. We call such a problem as *graph reconstruction under constraint G_c*. We will elaborate the detail of our solutions for this problem in Section IV. After reconstructing G_r from $\mathbf{d}'$ under constraint G_c, $G_r \cup G_c$ will be returned as the final result.

III. CONNECTIVITY SKELETON

In this section, we tackle the problem: constructing a robust connectivity skeleton by degree sequence $\mathbf{d}$, i.e., the first step of Algorithm 1. We start with a straightforward approach in Subsection III-A. Then, we present a more effective level-wise approach in Subsection III-B.

A. A baseline approach

The baseline method (shown in Algorithm 2) consists of two major steps. In the first step, we organize all vertices in $V_{\geq 2}$ (vertices with degree larger than 2) into a cycle. Then, we attach all vertices of degree 1 to those vertices in $V_{\geq 3}$ to hold more edges. In the above procedure, we use *avoiding disassortative mixing by degree* as the major heuristic to ensure the robustness of the resulting network. To implement this principle, we attach vertices with degree 1 to other vertices with non-zero residue degree in the ascending order of residue degree (lines 5-7).



(a) Cycle (b) Star-like graph

Fig. 2. Two graphs with the same size but different robustness

Algorithm 2 Naive Construction($\mathbf{d}$)

Input: $\mathbf{d}$: a graphical degree Sequence;
Output: A simple graph realizing $\mathbf{d}$;
{Step 1: Construct a cycle for $V_{\geq 2}$;}
Organize all vertices $V_{\geq 2}$ into a cycle;
For each $v \in V_{\geq 2}$, update its degree by $deg(v) - 2$;
{Step 2: Attach the vertices of degree 1};
$V_t \leftarrow V_{=1}$;
while $V_t \neq \emptyset$ **do**
  Select a vertex $v \in V_{\geq 3}$ with the minimum residue degree;
  Attach $deg(v)$ vertices of degree 1 to v, let V' be the set of all attached vertices;
  $deg(v) \leftarrow 0; V_t \leftarrow V_t - V'$;
end while
return the constructed graph;

Principle 1: Avoiding disassortative mixing by degree: Disassortative mixing by degree, that is vertices of small degree tend to connect to those with high degree, is one of the reasons for the frangibility of a network [15]. With the same number of vertices and edges, disassortative-mixing graphs (such as the star-like graph shown in Figure 2(b)) is clearly more fragile than that of assortative graphs (such as the cycle shown in Figure 2(a)).

B. A level-wise approach

The baseline approach can be further improved due to Lemma 2, which obviously holds true from problem definition. The lemma implies a level-wise construction of robust network. The basic idea is arranging vertices in a level-wise layout such that vertices in the lower levels always have degree smaller than those in upper levels. Simutaneously, we keep enough connectivity within each level and between adjacent levels so that the removal of upper level will not change the connectivity of lower levels.

Lemma 2: For a graph G under HDA, the maximum $f_\epsilon(G)$ is $(1-\epsilon)N$. The maximum is reached when G satisfies that: *for each attack in the first* $(1-\epsilon)N$ *iterations of HDA, the remaining network is connected.*

Given a *non-increasing* degree sequence $\mathbf{d}$ and corresponding vertex sequence V, we can partition V into disjoint classes: $\mathcal{P} = \{C_1, C_2, ..., C_k\}$ such that for any $1 \leq i < j \leq k$, degrees of vertices in C_i are larger than vertices in C_j. If we arrange such classes in a top-down fashion, classes with small subscripts lie in *upper levels* and those with large subscripts lie in *low levels*. Two classes C_i and C_{i+1} are *adjacent*.

1) Principles: Besides Principle 1, we also used another four principles to construct G_c in the heuristic approach:

Principle 2.Enough connectivity within each level: Consider the subgraph induced by vertices in the same level, i.e., $G[C_i]$ induced by level C_i. If $G[C_i]$ is quite sparse, connectivity of each $v \in C_i$ is dominated by its links to other levels. Once the edges between C_i and other levels are attacked, C_i will be detached from the graph. To avoid this, we organize vertices within each level as rings to ensure enough connectivity within each level.

Principle 3.Enough connectivity between adjacent levels: This is the most weak requirement, since the connectivity skeleton at least needs to be a connected graph. Following this rule, in the resulting skeleton, each level (for example C_i) except the first and last level is at least linked to two adjacent levels, i.e., C_{i-1} and C_{i+1}. As a result, under HDA, even when all vertices before level C_{i+1} are removed, the rest vertices still remain connected.

Principle 4.Link each vertex to its lower level: More formally, *each* $v \in G[C_i]$ *should be linked to at least one vertex in* $G[C_{i+1}]$. Consider the process of HDA, suppose v is the last vertex in $G[C_i]$ to be attacked. The links from v to $G[C_{i+1}]$ is the only path to maintain the connectivity of v to the remaining graph. Hence, this principle holds.

Principle 5.Homogenous link distribution: Given the same numbers of vertex and edges, homogenous degree distribution is more preferred to construct robust networks than the heterogenous ones. Consider the previous example shown in Figure 2. Another explanation of the fact that G_1 is more robust than G_2 is: G_1's links are distributed on vertices more equally than G_2's.

2) The algorithm: Based on above principles, we propose a level-wise construction algorithm for a connectivity skeleton. The procedure is shown in Algorithm 3. The algorithm consists of three major steps. First, we construct a partitioning on vertex set. Second, we link vertices within each level. Finally, add links between adjacent levels. Next, we elaborate each step.

Algorithm 3 Level-wise construction($\mathbf{d}, \gamma$)

Input: $\mathbf{d}$:Degree Sequence; $\mathcal{P}$: partition on vertex sequence; α: average degree of the subgraph induced by each level in $\mathcal{P}$
Output: Connectivity skeleton G;
{*Step 1: Construct partitioning on vertex set* $\mathcal{P} = \{C_1, C_2, ..., C_{m+2}\}$};
{*Step 2: linking vertex within each level*};
for all $C_i \in \mathcal{P}$ $(i \leq m+1)$ **do**
  Organize all vertex in C_i as a cycle;
end for
{*Step 3: linking vertex between adjacent levels;*}
for all level pair of $< C_i, C_{i+1} > (1 \leq i \leq m-2)$ **do**
  Link each vertex in C_i to some vertex of C_{i+1}
end for
Add $min\{|C_{m-1}|, |C_m|\}$ links between C_{m-1} and C_m;
Add $min\{|C_m|, |C_{m+1}| - |V_{=2}|\}$ links between C_m and C_{m+1};
Link vertex in C_{m+2} to vertex with non-zero residue degree from lower level to upper level;
return the constructed graph;

Step1: Partitioning vertex sequence: In the first step, we divide the vertex sequence in descending order of degree into a *geometric partitioning* $\mathcal{P} = \{C_1, C_2, ..., C_{m+2}\}$ such that:

- (1) $deg(u) \leq deg(v)$ if $u \in C_i, v \in C_j, i \leq j$;
- (2) $C_{m+1} = V_{=2} \cup V_{=3}$ and $C_{m+2} = V_{=1}$;
- (3) $|C_{i+1}| = \gamma |C_i|$ for each $1 \leq i \leq m-1$ and $|C_m| = |V_{\geq 4}| - |\frac{\gamma^{m-1}-1}{\gamma-1}|$, where $\gamma > 1$ is an integer.

In the partitioning, C_{m+1} contains all vertices with degree 2 or 3 and C_{m+2} contains all vertices with degree 1. Vertices in $V_{\geq 4}$ are partitioned into $C_1, ..., C_{m-1}$ such that $|C_{i+1}| = \gamma |C_i| (\gamma > 1)$ and C_m contains all the remaining vertices of $V_{\geq 4}$. This partitioning is called *geometric partitioning* because $|C_1|, |C_2|, ..., |C_{m-1}|$ is a geometric sequence. In general, we set $|C_1| = 1$ or some small integers, which has no influence on the result.

Step 2: Linking vertices within each level: In this step, we address Principle 2. We organize vertices in each C_i as a single cycle to ensure enough connectivity within C_i. In fact, we can organize C_i as a denser subgraph with more complex structure. However, we choose single cycle due to the following two reasons. First, cycle is the basic building block to ensure the robustness under HDA. Second, cycle consumes only two degrees for each vertex, which allows more vertices to have non-zero residue degrees and consequently it is more possible to precisely preserve the degree sequence.

Step 3: Linking vertices between adjacent levels: In this step, we address Principle 1 and 3-5. To ensure the connectivity between adjacent levels, say C_i and C_{i+1}, we link each vertex of C_i ($i \leq m-2$) to some vertex of C_{i+1} (line 4-6). After this step, each vertex in C_i ($1 < i \leq m-2$) have four edges, two within the level and two linking to the adjacent levels. Hence, in the partitioning step, we only construct geometric partitioning on $V_{\geq 4}$.

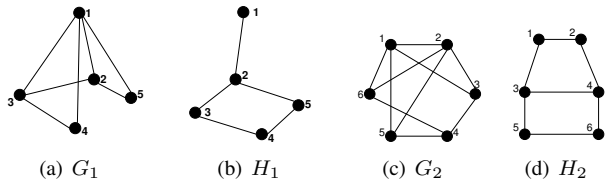
Next, we give the detail to handle three special cases when $i \geq m-1$, corresponding to line 8-10. In line 7, we add $min\{|C_{m-1}|, |C_m|\}$ edges between C_{m-1} and C_m since $|C_{m-1}|$ is not necessarily smaller than $|C_m|$. In line 8, we add $min\{|C_m|, |C_{m+1}| - |V_{=2}|\}$ edges between C_m and C_{m+1}. In C_{m+1}, vertices with degree 2 have no residue degrees and can not attract more links. Furthermore, $|C_{m+1}| - |V_{=2}|$ is not necessarily larger than $|C_m|$. Hence, we take the minimum of them as the link number. Finally, in line 9, we only need to attach vertices in C_{m+2}, i.e. vertices with degree 1 to the current network. In this step, following Principle 1, vertices with small degrees (in lower levels) has the highest priority to attract vertices of degree 1.

IV. Degree sequence matching

In this section, we solve *graph reconstruction under constraint G_c* problem. We first give the formalized model. Then, we give the algorithm to solve this problem.

A. Problem model

We formalize the *graph construction under constraint G_c* (GCC) problem in Definition 2. Disjointness with G_c and degree sequence matching are the two key issues when constructing G_r. Clearly, the disjoint constraint poses new challenges on degree sequence matching. When G_c is empty, this problem degrades to the traditional *graph reconstruction from a given degree sequence* problem. Hence, our problem is more general.



(a) G_1 (b) H_1 (c) G_2 (d) H_2

Fig. 3. Residue degree sequence

Definition 2 (GCC): Given a vertex set V, degree sequence $\mathbf{d}$ defined on V and a graph G_c on V, construct a graph G_r realizing $\mathbf{d}$ such that $E(G_r) \cap E(G_c) = \emptyset$.

Next, we show three facts about the residue degree sequence $\mathbf{d}'$ (produced in line 3 in Algorithm 1).

First, $\mathbf{d}'$ *is not necessarily in the non-increasing order of* $deg_G(v_i)$. This is not significant since our degree sequence matching solution (shown in Algorithm 4) is independent on degree order.

Second, $\mathbf{d}'$ *is not necessarily a graphical sequence any more*. As an example, consider G_1 and its skeleton H_1 in Figure 3, $\mathbf{d}(G_1)$ is $(4,3,3,2,2)$ (corresponding vertex sequence is (1,2,3,4,5)), and $\mathbf{d}(H_1)$ is $(1,3,2,2,2)$. Consequently, the residual sequence $\mathbf{d}'$ is $(3,0,1,0,0)$, which is obviously non-graphical.

Third, $\mathbf{d}'$ *is not not necessarily graphical under constraint G_c even if it is graphical on its own*. That is, any simple graph realizing $\mathbf{d}'$ contains at least one edge of G_c. For example, consider G_2 and its skeleton H_2 in Figure 3, the residue degree sequence by subtracting $\mathbf{d}(H_2)$ from $\mathbf{d}(G_2)$ is (2,2,0,0,1,1). It is graphical on its own, however, can not be realized without introducing edges that already exist in H_2. Non-graphicality of residue degree sequence demands approximate solutions.

Characterizing the graphicality of a degree sequence under a certain constraint in general is challenging. A both sufficient and necessary condition only exists for star constraint [16]. For more general constraints, such as an arbitrary connectivity skeleton G_c, finding a sufficient and necessary condition is an open problem to the best of our knowledge. Here, we give a sufficient condition in Theorem 1. Specifically, let G and H be two graphs with the same vertex set V. If for each vertex v, $deg_H(v) \leq deg_G(v)$, we say that *degree sequence of G subsumes that of H*, denoted by $\mathbf{d}(H) \leq \mathbf{d}(G)$. Then $\mathbf{d}_G - \mathbf{d}_H$ is graphical if one graph in $\mathcal{G}(\mathbf{d}_H)$ is a subgraph of $\mathcal{G}(\mathbf{d}_G)$, which is given in Theorem 1.

Theorem 1 (Sufficient condition): Let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two graphical degree sequences of V such that $\mathbf{d}_2$ is subsumed by $\mathbf{d}_1$. If $\exists G_2 \in \mathcal{G}(\mathbf{d}_2), \exists G_1 \in \mathcal{G}(\mathbf{d}_1)$ such that G_2 is a subgraph of G_1, then $\mathbf{d}_1 - \mathbf{d}_2$ is graphical.

B. The algorithm

We propose a greedy algorithm to solve GCC. The detail is shown in Algorithm 4. Given a vertex v_i, we imagine there are $deg(v_i)$ stubs anchored at the vertex, but the other ends are free. Connecting two stubs at two distinct nodes will form an edge. The algorithm processes each vertex v_i with

$deg_{\mathbf{d}'}(v_i) > 0$ unmatched stubs as follows: *connect v_i to the other $deg_{\mathbf{d}'}(v_i) > 0$ vertices that have the largest residual degrees and repeat until all vertices' stubs are matched or no other vertices have stubs to be matched*. In other words, we always select the vertex v with the largest $deg_{\mathbf{d}'}(v) > 0$ to connect (line 5). After the two stubs linked among v_i and v, their residue degrees are updated (line 10). There are two possible termination of the algorithm: all vertices stubs are connected (line 5), or no other no other vertices have stubs to be matched (line 6-8). Constraint imposed by G_c will prevent edges that already exist in the current network to be added (line 9).

In Algorithm 4, we first sort $\mathbf{d}'$ in non-increasing order of degree. In this way, we ensure that vertices with similar degree are connected to each other. As a result, we avoid the disassortative mixing, which is desired to construct robust network as required in Principle 1.

Algorithm 4 has time complexity $O(N+M)$, where $M = \sum deg_{\mathbf{d}'}(v_i)$. Because at most $\sum deg_{\mathbf{d}'}(v_i)$ pair of vertices will be linked.

Algorithm 4 DegreeMatching($\mathbf{d}'$,G_C)

Input: $\mathbf{d}'$:residue degree sequence, G_C: connectivity skeleton
Output: A graph G_r constructed from $\mathbf{d}$;

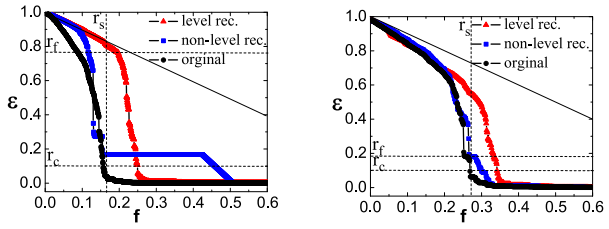
```
1: G_r ← ∅; i ← 0;
2: Sort d' in descending order of degree;
3: while deg_d'(v_i) > 0 do
4:    while deg_d'(v_i) > 0 do
5:       Select a vertex v ≠ v_i with the largest deg_d'(v);
6:       if deg_d'(v) = 0 then
7:          return G_r;
8:       end if
9:       if (v_i, v) ∉ G_c then
10:         G_r ← G_r ∪ {(v_i, v)}; deg_d'(v_i) − −; deg_d'(v) − −;
11:      end if
12:   end while
13:   i + +;
14: end while
15: return G_r;
```

V. Experimental Results

In this section, we conduct extensive experiments on both real networks and synthetic networks to show the effectiveness and efficiency of our solution. We first give the robustness metric to be explored. Then, give the detailed experimental results.

A. Metrics of robustness

To explore the robustness of a complex network under HDA, we need to observe the evolution of the size of the current MCC (i.e., ϵN with $1 < \epsilon < 1$) when the fraction of vertex removed (f) increases. Thus, under HDA, ϵ will be a decreasing function of f for any graph. Fast decrease of ϵ implies vulnerability of the network. If the network is a complete graph, which is the theoretically most robust network among networks with the same number of vertex, ϵ is linearly decreasing function of f, i.e., $\epsilon = 1 - f$. Hence, for a complete graph, its integral of $\epsilon - f$ curve is a triangle, whose area is the largest among all graphs with the same number of vertex.



(a) BA networks (b) ER networks

Fig. 4. Robustness of synthetic networks. The reconstructed networks optimized by level-wise approach are more robust than its original network and that produced by baseline approaches.

In general, the robustness of the network G can be measured by the area of integral of $\epsilon - f$ curve, that is:

$$\Delta(G) = \int_0^1 h(f)df \tag{1}$$

, where $\epsilon = h(f)$. Thus, larger $\Delta(G)$ implies a more robust network.

For a graph G, we will compare $\Delta(G)$ to $\Delta(G')$, where G' is the graph with robustness optimized. For the convenience of observation, we summarize the following statistics:

$$r_\Delta = \frac{\Delta(G') - \Delta(G)}{\Delta(G)} \tag{2}$$

, which is the relative improvement of the overall robustness;

$$r_S = \max\{|h_G(f) - h_{G'}(f)| : f \in (0,1)\} \tag{3}$$

, which is the largest improvement of ϵ for $f \in [0,1]$ (where $h_G(f)$ is the ϵ function for graph G);

$$r_f = \max\{|h_G^{-1}(\epsilon) - h_{G'}^{-1}(\epsilon)| : \epsilon \in (0,1)\} \tag{4}$$

, which is the largest extra cost needs to be paid by adversaries to obtain the same destructive effects by HDA.

We are also interested the *critical point* f_c, which is the fraction of vertex removed that will lead to the collapse of the structure of the network. In the following experiments, f_c will be identified by the fraction of vertex removed when the size of the remaining network is $\epsilon_c = 10\%$ of the original network.

B. Robustness

In this section, we will show the robustness of the network constructed by the level-wise construction (short for *'level rec.'*) with the comparison to the baseline approach (short for *'non-level rec.'*). Recall that the order of vertex in the residue degree sequence matching is critical for the robustness of the recontracted network. By default, descending order of degrees is used. We call such a strategy as *'preferentially linking'* (short for PL) since vertex of large degree tend to be linked to other vertex with large degree. Vertices may be processed in a random order, which is referred to as *'randomly linking'* (short for RL). For baseline approach, both two strategies are compared, denoted by 'Baseline-PL' and 'Baseline-RL', respectively. In all experiments, HDA is simulated by recalculating the largest degree of the remaining network.

1) Results on synthetic networks: We first generate synthetic graphs following two typical models: scale-free BA model producing power-law degree distribution and ER model which link each pair of vertex with probability p independently. The parameters of generated synthetic networks are shown in Table III.

The results on synthetic graphs are shown in Figure 4. In the plot, *'non-level rec.'* represents Baseline-PL since Baseline-RL shows similar results. The solid line is the $\epsilon - f$ plot of the complete graph with the same number of vertices. The dotted horizontal lines r_f denotes the value of ϵ when r_f is reached. r_c denotes $\epsilon = \epsilon_c$. The dotted vertical line r_S denotes the value of f when r_s is reached. All notations in the results of following experiments have the same meanings without extra statement.

Robustness of networks produced by level-wise approach: As shown in Figure 4, for both BA and ER models, networks reconstructed by level-wise approach are more robust than its original one. Specifically, f_c of the original network increases from 15.7% to 25% and from 26.9% to 34.3%, for BA network and ER network, respectively, which is a significant improvement of network robustness. For both two networks, before a certain f (for BA $f = 0.25$, for ER $f = 0.35$), reconstructed network consistently exhibits larger ϵ than its original version, indicating that at the initial stage of attack, reconstructed network shows better robustness than its original version. This is important since in real attacks it is quite possible that adversaries has only limited resource to issue first-stage attacks. We also can see that the reconstructed network produced by level-wise approach is close to the upper bound of the theoretically most robust network (solid line in the plot) when f is small, especially on BA networks.

More evidences are provided by r_f, r_S, r_Δ and f_c in Table II. Consider statistics of level-wise approach on BA networks. In the best case, extra $r_f = 10.9\%$ vertices need to be removed to achieve the same destructive effect in our reconstructed network. Large r_s (close to 78%) suggests that in the best case the integrity of network can be significantly improved in the reconstructed network. $\Delta f_c = 9.3\%$ (which is $f_c' - f_c$) implies that extra 300 vertices need to attacked to malfunction the network. Compare $\Delta H_{G'}$ and ΔH_G. r_Δ reaches to 70.9%, indicating a significant improvement of overall robustness performance. All these facts suggest that the improved robustness of the reconstructed network produced by level-wise approach is quite significant. Similar results can be observed for ER networks.

Comparison of different approaches: Next, we compare the robustness of networks constructed by different approaches. From Figure 4, we can see that the optimized network constructed by baseline approach can maintain the integrity of the network only at the initial stage of the attack. When the attacking continues, the network tends to collapse fast. Compared to the baseline approach, the level-wise approach produces graphs with more consistent resilience to HDA. From Table II we also can see that for both BA and ER networks, $\Delta(G')$ and f_c' achieved by level-wise approach is significantly larger than that of Baseline-PL and Baseline-RL.

2) Results on real networks: We use real networks to show the effectiveness of our approach. Three of them are infrastructure networks, including UsPowerGrid [36]: the power grid network of Western States in the US; USAirline: airline transporting networks of US [35] and InternetAs: Internet at autonomous level [10]. The fourth network are protein-protein interaction (PPI) network of Yeast [29] (denoted by Yeast), whose robustness accounts for the function of PPI networks. The basic statistics of these networks are shown in Table III.

The results of our reconstructed networks from these real networks are shown in Figure 5 and Table II. It is quite impressive to see that all these real networks, especially infrastructure networks, are quite sensitive to HDA; and after optimization, the reconstructed networks exhibit significantly more resilience to HDA compared to the original network.

As an example, we highlight that InternetAS is quite fragile under HDA and its robustness can be significantly improved by our approach. On this network, only $f_c = 2.26\%$ vertices of highest degree need to be removed to disintegrate the network. However, after reconstruction, f_c is increased to 8.6%, which implies that extra 6.3% vertices need to be removed to fragment the network into pieces. In the worst case, to obtain the same destructive effect, the adversaries need to pay extra $r_f = 7.8\%$ cost (in Table II). It is also quite surprising to see that along the attacking process, r_S can reach to 71.4%, which implies that by attacking the same number of targets of highest degrees integrity of the reconstructed network can be significantly improved than that of the original one. The improvement of overall robustness performance is also quite impressive, since r_Δ reaches to 534.9%.

Compared to the baseline approach, for all real networks, level-wise approach exhibits its advantage. As shown in Table II for all real networks, $\Delta(G')$ and f_c' achieved by level-wise approach is significantly large than that of Baseline-PL and Baseline-RL.

C. Preservation of degree sequence

In this subsection we will show that the reconstructed network preserves the degree sequence of the original one. We summarize Err_n, Err_e between the original network and the reconstructed networks in Table III.

As shown in Table III, the error rate whatever measured by Err_n or Err_e is quite small and in most cases is close or equal to 0. For example, for BA, UsPowerGid and UsAirline, the reconstructed network strictly preserves the degree sequence of the original network, yielding $Err_n = Err_e = 0$. Such facts are quite surprising since significant improvement of robustness of these networks has been achieved.

Consider those networks with $Err_n \neq 0$. The size of reconstruct networks produced by our approach does not exceed the size of the original one. Hence, our approach uses less edges to achieve more robustness. For example, for InternetAs, the reconstructed network use less edges (about 8.38%, see the edge numbers in Table III) to achieve significant robustness improvement ($\Delta f_c = 6.3\%$). This fact sufficiently shows that our approach is quite effective in the sense of organizing the vertex and edges of a graph into a robust network.

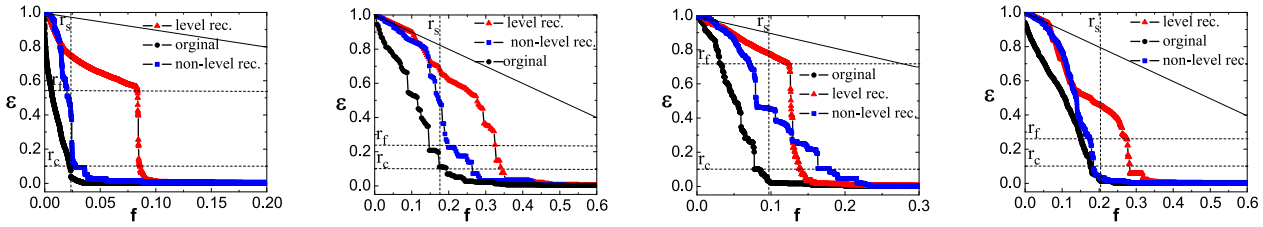
Compared to the baseline approaches, level-wise approach win in all cases except InternetAs (when compared to Baseline-PL), which shows again the advantage of level-wise approach

TABLE II. IMPROVEMENT OF ROBUSTNESS OF RECONSTRUCTED NETWORK.

	Original Network		Level-wise reconstruction						Baseline-RL		Baseline-PL	
Network	$\Delta(G)$	f_c	$\Delta(G')$	r_Δ	r_s	r_f	f'_c	Δf_c	$\Delta(G')$	f'_c	$\Delta(G')$	f'_c
BA	0.118	0.157	0.202	0.709	0.780	0.109	0.250	0.093	0.177	0.459	0.143	0.280
ER	0.201	0.269	0.243	0.207	0.489	0.08	0.343	0.074	0.213	0.300	0.191	0.284
InternetAS	0.0095	0.0226	0.060	5.349	0.714	0.078	0.086	0.063	0.022	0.036	0.036	0.068
UsPowerGrid	0.052	0.084	0.113	1.173	0.752	0.093	0.140	0.056	0.103	0.179	0.073	0.094
UsAirLine	0.110	0.194	0.240	1.169	0.575	0.180	0.342	0.148	0.174	0.264	0.213	0.316
Yeast	0.099	0.179	0.180	0.818	0.443	0.125	0.283	0.104	0.133	0.184	0.162	0.242

TABLE III. STATISTICS OF NETWORKS.N:#VERTEX, M:#EDGE; $Z = 2M/N$; Err_n, Err_e ARE AT THE MAGNITUDE OF 10^{-4}.

	Original Network			Level-wise reconstruction			Baseline-PL			Baseline-RL		
Network	N	M	Z	M	Err_n	Err_e	M	Err_n	Err_e	M	Err_n	Err_e
BA	3000	5980	3.992	5980	0	0	5980	0	0	5980	0	0
ER	1000	1995	3.99	1994	10	2.51	1994	0	0	1995	10	2.51
InternetAS	22442	45550	4.06	41710	12.9	838	43844	2.67	374	39008	20.5	1312
UsPowerGrid	4941	6594	2.669	6594	0	0	6594	0	0	6591	4.05	9.09
UsAirline	332	2126	12.81	2126	0	0	2125	301	4.7	1901	90.4	1058
yeast	2361	7182	6.083	6913	4.24	0.723	6913	8.47	1.446	6880	29.64	47.73



(a) InternetAS (b) USAirline (c) USPowerGrid (d) Yeast

Fig. 5. Robustness on real networks. Our approach significantly improves the robustness of real networks.

in degree-preservation. The comparison between Baseline-PL and Baseline-RL shows that networks produced by Baseline-PL can preserve more degrees than Baseline-RL.

VI. RELATED WORKS

Our works is closely related to the following research:

Attack model: Understanding the attacking model is crucial for the success of protect network. Vertex importance measure is important for the selection of the attack target. Degree, betweeness centrality [8], centrality based on random walks [23], PageRank [24], HITS [17], shell index defined by k-core decomposition [20] and node abnormality [28] are potential measures can be used to select victim. Among them, degree can easily computed, and most other measures are computational costly. Hence, degree is widely used in practical evil attack, and HDA becomes the most popular attack strategy. Another aspect to characterize an attack model is the evaluation of network robustness. The maximum connected component size is the one of widely used measures of network robustness [1]. Lin. et.al [19] provided a survivability mechanism called Degree of Disconnectivity (DOD) for the network operator to detect risks. Tong et.al [33] uses the first eigenvalue of the adjacency matrix to measure the 'vulnerability' of a network. All these works are related to our research. But they do not investigate how to optimize the robustness of a given network.

Network robustness optimization: Quite recently, optimization of robustness of complex networks began to attract research interests. Some works focus on seeking an appropriate degree distribution to optimize network robustness without changing network size and density. Tanizawa et.al. optimize network robustness by designing a bimodal degree distribution against waves of targeted and random attacks [30], and further find optimal parameter of multimodal degree distribution to realize a robust network [31]. Paul et.al. [25] find optimal parameters of power-law degree distribution so that robustness of these networks against both random failure and intentional attack is optimized. Moreira et.al [21] propose a failure model in which a topological bias can be tuned and influence the network robustness under the model. Beygelzimer et.al [2] empirically study the influence of different edge rewiring strategies on the improvement network robustness. Herrmann et.al. [14] proposed a greedy solution, which randomly select an edge pair and swap them if the robustness can be improved by this swapping.

However, most these works do not address the degree sequence preservation issue. The greedy solution proposed in [14] can preserve the degree sequence. But it has the following weakness. First, no mechanisms or principles to contract a robust network are proposed. Second, it is inefficient. It needs to swap almost 20% of all edges to get desired robustness improvement. Each selection of an edge pair implies one time computation of network robustness metrics. It use the average

component size used in [14], which costs exactly linear time. Hence, in general, the edge-swapping solution is not efficient.

VII. Conclusion

In this paper, we investigate the problem to reconstruct a network maximizing robustness under the degree sequence constraint. We developed the algorithmic solutions for this problem. We propose a level-wise approach to construct a *connectivity skeleton* accounting for the robustness of the reconstructed network. We propose a new problem *graph reconstruction under constraint* to satisfy the residue degree sequence and develop a greedy algorithm to solve this problem. We conduct extensive experiments to show that our solution can significantly improve the robustness of a complex networks while pressing the degree constraint.

VIII. Acknowledgments

This work was partially supported by Natural Science Foundation of Shanghai under Grant No. 13ZR1417700, by 085-Energy Economy and Service Open Project, by the National NSFC (No.61472085, 61171132, 61033010), by National Key Basic Research Program of China under No.2015CB358800, by Shanghai STCF under No.13511505302, by NSF of Jiangsu Prov. under No. BK2010280, by the NSF under Grant No. IIS1017415, by the Army Research Laboratory underx Cooperative Agreement No. W911NF-09-2-0053, by Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-11-C-0200 and W911NF-12-C-0028, and by Region II University Transportation Center under the project No. 49997- 33 25.

References

[1] R. Albert et al. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[2] A. Beygelzimer et al. Improving network robustness by edge modification. *Physica A*, 357:593–612, 2005.

[3] L. Briesemeister et al. Epidemic profiles and defense of scale-free networks. In *WORM '03*.

[4] R. A. Brualdi. Matrices of zeros and ones with fixed row and column sum vectors. *Linear Algebra and its Applications*, 33(1):159–231, 1980.

[5] D. Chakrabarti et al. Epidemic thresholds in real networks. *ACM Trans. Inf. Syst. Secur.*, 10.

[6] R. Cohen et al. Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, 86(16):3682–3685, 2001.

[7] P. Erdős et al. Graphs with prescribed degrees of vertices (hungarian). *Mat. Lapok*, 11:264–274, 1960.

[8] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[9] A. Ganesh et al. The effect of network topology on the spread of epidemics. In *INFOCOM'05*.

[10] T. C. Group. The caida as relationships dataset, 2003-2007, http://www.caida.org/data/active/as-relationships/.

[11] D. Gruhl et al. Information diffusion through blogspace. In *WWW'04*.

[12] S. Hakimi. On the realizability of a set of integers as degrees of the vertices of a simple graph. *J. SIAM Appl. Math.*, 10:496–506, 1962.

[13] V. Havel. A remark on the existence of finite graphs. *Casopis Pest. Mat.*, 80:477–480, 1955.

[14] H. J. Herrmann et al. Onion-like network topology enhances robustness against malicious attacks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(01):P01027, 2011.

[15] P. Holme et al. Attack vulnerability of complex networks. *Phys. Rev. E*, 65(5):056109, 2002.

[16] H. Kim et al. Degree-based graph construction. *Journal of Physics A*, 42(39):392001, 2009.

[17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46.

[18] A. Krause et al. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

[19] F. Y.-S. Lin et al. Maximization of network survivability considering degree of disconnectivity. In *ICCSA'11*.

[20] J. Moody et al. Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups. *American Sociological Review*, 68(1):103–127, 2003.

[21] A. A. Moreira et al. How to make a fragile network robust and vice versa. *Phys. Rev. Lett.*, 102(1):018701, 2009.

[22] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[23] M. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39 – 54, 2005.

[24] L. Page et al. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, November 1999.

[25] G. Paul et al. Optimization of robustness of complex networks. *Eur. Phys. J. B*, 38(2):187 – 191, 2004.

[26] R. Roman et al. On the security of wireless sensor networks. In *ICCSA'05*.

[27] R. V. Solé et al. Robustness of the european power grids under intentional attack. *Phys. Rev. E*, 77(2):026102, 2008.

[28] J. Sun et al. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM'05*.

[29] S. Sun et al. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003.

[30] T. Tanizawa et al. Optimization of network robustness to waves of targeted and random attacks. *Phys. Rev. E*, 71(4):047101, 2005.

[31] T. Tanizawa et al. Optimization of the robustness of multimodal networks. *Phys. Rev. E*, 74(1):016125, 2006.

[32] R. Taylor. Switchings constrained to 2-connectivity in simple graphs. *SIAM. J. on Algebraic and Discrete Methods*, 3(1):114–121, 1982.

[33] H. Tong et al. On the vulnerability of large graphs. In *ICDM'10*.

[34] Y. Tu. How robust is the internet? *Nature*, 406(6794):353–354, 2000.

[35] S. V. Ukkusuri et al. Robust transportation network design under demand uncertainty. *Computer Aided Civil and Infrastructure Engineering*, 22(1):6–18, 2007.

[36] D. J. Watts et al. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.