Diversity and Trustedness to Increase Structural Robustness in Networks

Waseem Abbas, Aron Laszka, and Xenofon Koutsoukos

Accepted for publication in the proceedings of the 2019 American Control Conference

***Abstract*— In a networked system, any change in the underlying network structure, such as node and link removals due to an attack, could severely affect the overall system behavior. Typically, by adding more links and connections between nodes, networks can be made structurally robust. However, this approach is not always feasible, especially in sparse networks. In this paper, we aim to improve the structural robustness in networks using the notions of *diversity* and *trustedness*. Diversity means that nodes in a network are of different types and have many variants. Trustedness means that a small subset of nodes are immune to failures and attacks. We show that by combining diversity and trustedness within the network, we can significantly limit the attacker's ability to change the underlying network structure by strategically removing nodes. Using pairwise connectivity as a measure, we show that by appropriately distributing trusted nodes and assigning types to nodes, network robustness can be significantly improved. We analyze the complexity of diversifying and computing a set of trusted nodes, and then present heuristics to compute attacks consisting of node removals. We also present heuristics to defend networks against such attacks by distributing node types and trusted nodes. Finally, we evaluate our results on a set of benchmark networks to demonstrate the usefulness of our approach.**

***Index Terms*— Pairwise connectivity, network robustness, network structure, diversity, trusted nodes.**

I. Introduction

In networked systems, the overall dynamics at the system level are deeply influenced by the underlying network structure. Any change in the network topology, such as failure of components or loss of interconnections between components could significantly alter the overall system behavior. Such events could make the networks disconnected or significantly reduce their connectivity, which is a key requirement for various distributed control algorithms, for instance to solve distributed consensus, or distributed estimation problems in the presence of adversaries (e.g., [1], [2]). The problem of characterizing and exploring ways to improve networks' structural robustness is of immense concern across various domains that benefit from the networking paradigm; for instance, networked cyberphysical systems including power grids, road networks, and water networks. In this paper, we study distinctive ways of improving network's structural robustness in the face of attacks consisting of strategic removals of nodes from the network.

W. Abbas is with the Electrical Engineering Department at the Information Technology University, Lahore, Pakistan. (email: w.abbas@itu.edu.pk)

A. Laszka is with the Computer Science Department at the University of Houston, TX, USA (email: alaszka@uh.edu).

X. Koutsoukos is with the Department of Electrical Engineering and Computer Science at the Vanderbilt University, Nashville, TN, USA (email: xenofon.koutsoukos@vanderbilt.edu).

To quantify the consequences of node (edge) removals, and characterize the robustness of network structure, numerous measures have been identified in the literature, including vertex (edge) connectivity, toughness [3], tenacity [4], fragmentability [5], r-robustness [1], and others (e.g., [6], [7]). These measures aim to quantify both the effort required to cause the damage; such as the number of nodes (edges) that can be removed; as well as the extent of damage, such as the number or sizes of components into which the network is fragmented. Here, we consider a widely used notion of *pairwise connectivity* to measure the structural robustness. Pairwise connectivity measures the overall fraction of node-pairs that remain connected through a path after a removal of nodes (edges) [8], [9], [10]. Since nodes belonging to the same component in a graph remain connected, pairwise connectivity, unlike the simple notion of network connectivity, assimilates both the number and size of components in the residual graph (e.g., see [9]). This feature makes pairwise connectivity particularly useful to understand vulnerability of communication networks to disasters and failures [11], [12]. In this article, we aim to develop approaches to improve the pairwise connectivity of the graph after an attack consisting of strategic removal of nodes.

Networks are made structurally robust typically by adding more links (edges) in the underlying graph, which we call a *redundancy* approach. Although effective, this redundancy approach is not always feasible, especially when networks are intrinsically sparse. So, we ask the question, *how to make sparse networks structurally robust without adding extra links between nodes?* Here, we propose to combine the notions of *diversity* and *trustedness* as an alternative to the redundancy approach to improve network's structural robustness, as measured by the pairwise connectivity. Diversity means that network components have multiple variants, and trustedness means that some of the nodes in the network are hardened and are immune to attacks. We discuss these notions in more detail in Section II. Both of these notions basically restrict the attackers choice to remove nodes, and the effect of this appears in the form of increased network robustness.

Our main contributions in the paper include combining the notions of diversity and trustedness (as explained in Section II) to improve the pairwise connectivity in networks. In this direction, first, we provide a network and attack models, and formulate the attacker and defender problems in the presence of diverse and trusted nodes (Section III). We then study the complexity of the defender problem, that is how to manage diversity and trustedness in the network to maximize its structural robustness, showing that the defender's problem is NP-hard (Section IV). We then provide heuristics to solve the

attacker as well as the defender problems and discuss them in detail (Section V). We then evaluate our approach and results using benchmark networks (Section VI). Finally, we present our conclusions and some future directions (Section VII).

II. Techniques to Improve Robustness

Here, we discuss various techniques through which structural robustness in networks can be improved. The common point in all such mechanisms is to somehow impede the attackers ability to launch an attack that can significantly reduce connectivity between nodes. These defense mechanisms can be categorized into *diversity*, *hardening*, and *redundancy*.

A. Redundancy

A typical approach to improving network's structural robustness is through redundancy, that is by having extra devices and extra connections between them than required. The basic idea is that in the case of removal or failure of some components, redundant devices will take over to ensure the smooth operations. In the context of improving connectivity between various components, extra links are typically created. If the network is modeled by a graph, it corresponds to including extra edges between nodes. The problem of strategically adding a minimum number of extra edges to improve graph connectivity is referred to as the *connectivity augmentation* problem, and has been studied for decades, for instance see [13], [14], [15], [16], [17]. Although effective, improving connectivity between various network components using redundancy approach is not always feasible or economical. Thus, we aim to utilize alternative approaches to improve network's structural robustness.

B. Diversity

To limit attacker's ability to compromise a system, one strategy is to *diversify* the individual components or devices, that is, have many variants of the same device. Owing to distinct implementations of such variants, they typically have disjoint exploitation sets and vulnerabilities. As a result, an attacker cannot compromise all the devices by exploiting a particular vulnerability. In fact, an attacker can only compromise devices belonging to the same type or class by exploiting a particular vulnerability specific to that class. The effectiveness of having heterogeneous implementations of devices towards increasing network robustness has been studied previously (e.g., see [8], [18], [19], [20], [21], [22]). Diversification of system components and devices can be achieved by employing different operating systems, software packages, and hardware platforms. The number of variants available for a particular application are usually limited. A random assignment of these variants to the components within the network is generally not a useful approach. In fact, a careful assignment of these variants to the network components is needed to improve network robustness using the diversification approach.

C. Trustedness

Another way to restrict an attackers' scope of action is by limiting the number of devices that can be compromised. This objective can be achieved by *hardening* a small subset of devices and making them trusted. It means that we can assume that such a small group of devices, called *trusted nodes*, are secured and will continue to operate normally. Adversaries cannot take control of them and system administrators (defenders) can rely on their operational integrity. As a result, an attacker can only compromise the non-trusted devices. Since hardening can be substantially expensive, only a small subset of devices can be hardened, and the defender needs to select them carefully. Previously, we studied the significance of trusted nodes towards improving network robustness in [23], [24], [25]. A combination of trustedness and diversification of nodes, however, substantially improves network robustness as compared to individual application of these techniques.

III. Network Model and Problem Formulation

(a) Network Graph: We model the network as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ is the set of *nodes* (such as routing nodes, clients, devices) and $\mathcal{E}$ is the set of *edges* representing communications and interactions between nodes. We assume that the network structure remains fixed based on the constraints of the networking application. An edge between two nodes u and v is denoted by an unordered pair (u, v). The neighborhood of node v, denoted by $\mathcal{N}(v)$, is the set $\{u : (u, v) \in \mathcal{E}\}$. Any two nodes interact with each other as long as they are connected, that is, there exists a *path* between them. We use the terms nodes and vertices interchangeably.

(b) Trusted nodes: These are the nodes which function correctly throughout the network lifetime, that is, they are hardened and are immune to attacks and failures. Consequently, we assume that these nodes remain operational at all times without faults and failures. We denote the set of trusted nodes by $\mathcal{T} \subset \mathcal{V}$.

(c) Diversity of nodes: We assume that nodes are heterogeneous, that is, there are multiple types (or variants) of nodes. The node types are denoted by the set $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_d\}$, and each node u belongs to one of the types in $\mathcal{D}$, which is assigned by the function Γ.

$$\Gamma : \mathcal{V} \longrightarrow \mathcal{D} \tag{1}$$

We denote the set of nodes of type $\mathcal{D}_i$ by $\mathcal{V}_i \subseteq \mathcal{V}$, and define $n_i = |\mathcal{V}_i|$.

(d) Attack: An attacker compromises a subset of nodes and remove them from the network, thus altering the structure, and hence the overall connectivity of the remaining network. An attack satisfies the following two constraints:

1) Nodes removed from the network as a result of an attack should belong to the same type.
2) A trusted node cannot be removed.

The set of nodes removed by an attacker are denoted by $\mathcal{A}_i \subseteq \mathcal{V}_i$, where the subscript i indicates that the removed nodes are of type $\mathcal{D}_i$.

(e) Structural Robustness Measure: As discussed earlier, there could be several measures of structural robustness. As a result of node removals, a network can be fragmented into multiple components. The connectivity between nodes in the remaining network depends both on the number and size of the resulting components. To deal with this issue, we consider *pairwise connectivity* as a measure of structural robustness of the network. Pairwise connectivity is simply the number of node pairs that are left connected in the residual graph as a result of a removal of some nodes from the original graph. If $|\mathcal{V}| = n$, and $|\mathcal{A}_i| = a$ nodes are removed, then the number of node pairs in the residual graph is $\binom{n-a}{2}$. Similarly, if $\mathcal{C}_j$ is the set of nodes in the j^{th} component of the residual graph, then all the node pairs in that component are connected. Thus, we define the pairwise connectivity $\mathcal{P}$ of the residual graph as

$$\mathcal{P} = \frac{\sum_{\mathcal{C}_j} \binom{|\mathcal{C}_j|}{2}}{\binom{n-a}{2}}. \tag{2}$$

where $|\mathcal{C}_j|$ is the number of nodes in the j^{th} component. Note that $\mathcal{P}$ is simply the fraction of node pairs that are connected through a path in the residual graph.

A. Problem Formulation

Our main objective is to minimize the damage – in terms of reducing the pairwise connectivity – as a result of an attack. Thus, we would like to increase the structural robustness of the network as measured by the pairwise connectivity. Instead of strategically adding edges to the graph, which is a typical way to increasing structural robustness in graphs, we resort to employing the node diversity and trusted nodes to increase the pairwise connectivity and minimize the damage inflicted by the attacker.

Here, we fix the attacker's budget, that is, the maximum number of nodes that can be removed is bounded, $|\mathcal{A}_i| \leq a$. Given the type assignment to nodes Γ, and the set of trusted nodes $\mathcal{T}$, the *attacker's goal* is to select the type $\mathcal{D}_i$, $i \in \{1, 2, \cdots, d\}$ and at most a nodes of that type to remove from the graph to minimize the pairwise connectivity, that is,

$$\arg\min_{i} \left(\arg\min_{\substack{\mathcal{A}_i \subset \mathcal{V}_i; \\ |\mathcal{A}_i| \leq a}} \sum_{\mathcal{C}_j} \binom{|\mathcal{C}_j|}{2} \right), \tag{3}$$

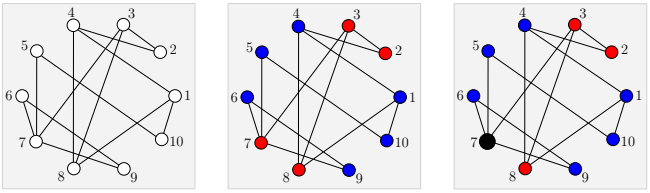
where $\mathcal{C}_j$ is the j^{th} component of the residual graph as a result of an attack $\mathcal{A}_i$. The *defender's objective* is to select a set of t trusted nodes $\mathcal{T} \subset \mathcal{V}$ and assign a type to each node in $\mathcal{V} \setminus \mathcal{T}$ from the set $\mathcal{D}$, that is compute Γ, such that the pairwise connectivity of the residual graph obtained after an optimal attack is maximized, that is

$$\arg\max_{\mathcal{T}, \Gamma} \left(\arg\min_{i} \left(\arg\min_{\substack{\mathcal{A}_i \subset \mathcal{V}_i; \\ |\mathcal{A}_i| \leq a}} \sum_{|\mathcal{C}_j|} \binom{|\mathcal{C}_j|}{2} \right) \right), \tag{4}$$

subject to $|\mathcal{T}| \leq t$ and $|\mathcal{D}| \leq d$.

We note that there are two sub-problems here – first, computing an optimal attack given the trusted nodes and node types; and second, selecting an optimal type assignment and trusted nodes to maximize $\mathcal{P}$ as a result of an optimal attack. A special case of the attacker problem, in which $\mathcal{T} = \emptyset$, and all nodes are of the same type $|\mathcal{D}| = 1$, is often referred to as the *critical node detection* problem and has been widely studied (e.g., see [26], [27], [28], [29]). It has been shown that computing a set of critical nodes, whose removal minimizes the pairwise connectivity of the remaining graph, is NP-complete [27] even for the simple case of $\mathcal{T} = \emptyset$ and $|\mathcal{D}| = 1$.

Example: To illustrate the problem, consider the graph in Figure 1(a), and attack consisting of removing two nodes. If all nodes are of the same type, then the optimal attack consists of removing nodes 1 and 7, and the pairwise connectivity of the remaining graph is 0.2857. If $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$, with $\mathcal{V}_1 = \{1, 4, 5, 6, 9, 10\}$ and $\mathcal{V}_2 = \{2, 3, 7, 8\}$ as shown in Figure 1(b). In this case, the optimal attack consists of removing nodes 2 and 7, and the resulting pairwise connectivity is 0.5714. Moreover, if there is a single trusted node, $\mathcal{T} = \{7\}$ as in Figure 1(c), then the optimal attack consists of nodes 1 and 5, and the resulting pairwise connectivity is 0.75, which is significantly better than the case in which there is no trusted node, and all the nodes belong to the same type.



(a) (b) (c)

Fig. 1: (a) A network with all nodes of the same type. (b) Each node belongs to one of the two types. (c) A single trusted node along with two types of non-trusted nodes.

IV. Computational Complexity

In this section, we show that computing an optimal set of trusted nodes and node types is a computationally challenging problem. More precisely, we show that the network resilience maximization problem, as defined below, is NP-hard.

Definition *(Network Robustness Maximization Problem (Decision Version))* Given a network graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a budget on the number of nodes that can be trusted t, a set of node types $\mathcal{D}$, an attacker budget a, and a threshold robustness

$\mathcal{P}^*$, find a set of trusted nodes $\mathcal{T} \subseteq \mathcal{V}$ and a node type assignment Γ such that $|\mathcal{T}| \leq t$ and

$$\mathcal{P}^* \leq \mathcal{P}(\mathcal{T}, \Gamma) = \arg\min_i \left(\arg\min_{\substack{\mathcal{A}_i \subset \mathcal{V}_i; \\ |\mathcal{A}_i| \leq a}} \sum_{\mathcal{C}_j} \binom{|\mathcal{C}_j|}{2} \right). \quad (5)$$

Theorem 4.1: The Network Robustness Maximization Problem is NP-hard.

We prove computational hardness by reducing a well-known NP-hard problem, the Set Cover Problem, to the Network Robustness Maximization Problem.

Definition Given a set U, a set $\mathcal{F}$ of subsets of U, and a threshold k, find a subset $\mathcal{C} \subseteq \mathcal{F}$ consisting of at most k subsets such that $\mathcal{C}$ covers U (i.e., for every $u \in U$, there exists a $C \in \mathcal{C}$ such that $u \in C$).

Proof of Theorem 4.1 – We reduce an instance $(U, \mathcal{F}, k)$ of the Set Cover Problem (SCP) to an instance $(\mathcal{G}, t, \mathcal{D}, a, \mathcal{P}^*)$ of the Network Robustness Maximization Problem (NRMP) as follows.

- Let $a = |\mathcal{F}|$.
- Construct a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as follows:
  - create a clique X of $a+1$ nodes;
  - for each element $u \in U$, create a clique Y_u of $a+1$ nodes;
  - for each set $C \in \mathcal{F}$, create a node v_C, and create edges between v_C and each node in X;
  - for each $u \in C$, $C \in \mathcal{F}$, create edges between v_C and each node in Y_u.
- Let $\mathcal{D} = \{\mathcal{D}_1\}$ (i.e., single node type).
- Let $t = k$ and $\mathcal{P}^* = (|\mathcal{V}| - a)^2$.

Clearly, the above reduction can be performed in a polynomial number of steps. It remains to show that the constructed instance of the NRMP has a solution if and only if the SCP instance has a solution. Note that since there is a single node type $\mathcal{D}_1 \in \mathcal{D}$, the node type assignment is simply $\Gamma(\mathcal{D}_1) = \mathcal{V}$ (i.e., $\mathcal{V}_1 = \mathcal{V}$). For this reason, we will disregard Γ for the remainder of this proof and consider only $\mathcal{T}$.

First, assume that the SCP instance has a solution $\mathcal{C}^*$. We show that $\mathcal{T}^* = \{v_C \,|\, C \in \mathcal{C}^*\}$ is then a solution to the NRMP. Clearly, $|\mathcal{T}^*| \leq t$ since $t = k$ and $|\mathcal{C}^*| \leq k$. Since $\mathcal{C}^*$ is a set cover, there exists for every clique Y_u at least one node $v_C \in \mathcal{T}^*$ such that v_C is connected to Y_u. Since the adversary cannot remove a node v_C that is in $\mathcal{T}^*$, every node from Y_u will remain connected to at least one node v_C after any attack. Further, since $|X| > a$, there will always remain at least one node from X after any attack, which connects all the nodes $v_C \in \mathcal{T}^*$ together. Therefore, the graph will remain connected after any attack. The pairwise connectivity of a connected graph with $|\mathcal{V}| - a$ nodes is $(|\mathcal{V}| - a)^2 = \mathcal{P}^*$, which proves that $\mathcal{T}^*$ is a solution to NRMP.

Second, assume that the SCP instance does not have a solution $\mathcal{C}^*$. We then show that for any set $\mathcal{T}^*$ of at most t trustworthy nodes, there exists an attack after which pairwise connectivity will be lower than $\mathcal{P}^*$. Since there is no set cover of at most $k = t$ subsets in the SCP instance, the subsets corresponding to the nodes in $\mathcal{T}^*$ cannot cover all the elements of U. Let $u \in U$ be an element that is not covered, and consider an attack that removes all non-trusted nodes $v_C \notin \mathcal{T}^*$ that correspond to the subsets $C \in \mathcal{F}$. Clearly, this attack is feasible since $|\mathcal{F}| = a$, and it will separate the clique Y_u from the remainder of the graph. Since the pairwise connectivity of any disconnected graph of $|\mathcal{V}| - a$ is lower than $(|\mathcal{V}| - a)^2 = \mathcal{P}^*$, we have that $\mathcal{T}^*$ cannot be a solution to NRMP, which concludes our proof. ∎

V. Local Search Metaheuristics for the Attacker and the Defender

Since both the attacker and the defender problems are NP-hard, we present algorithms to find an attack and a defense strategy based on a local search metaheuristic approach. In particular, we use *simulated annealing* to find a near-optimal attack $\mathcal{A}$, and defense strategy, that is $\mathcal{T}$ and Γ. In the next section, we present a numerical evaluation of these heuristics.

For the attacker case, the basic idea is to initially select an attack $\mathcal{A}_i \subset \mathcal{V}_i$ consisting of a nodes of the same type $\mathcal{D}_i$ arbitrarily, and then improve the quality of the attack in successive iterations. In each iteration, we explore a new attack $\mathcal{A}'_i$ by perturbing $\mathcal{A}_i$. If the pairwise connectivity of the residual graph as a result of $\mathcal{A}'_i$ is lesser as compared to $\mathcal{A}_i$, then $\mathcal{A}'_i$ replaces the previous solution. However, $\mathcal{A}'_i$ replaces the previous attack with a small probability even if it results in a higher pairwise connectivity. This probability is a function of the difference between the two solutions and a parameter often referred to as the 'temperature', which decreases exponentially with the number of iterations. In the perturbation step (line 7 in Algorithm 1), we randomly select a node $u \in \mathcal{A}_i$ and replace it with another randomly selected node $v \in \mathcal{V}_i \setminus \mathcal{A}_i$.

Algorithm 1 Simulated Annealing Algorithm for the Attacker

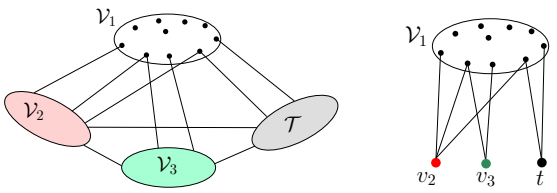
1: **Input** $\mathcal{G}$, $\mathcal{V}_i$, a, iterations.
2: **Output** $\mathcal{A}_i$
3: **Initialize:** $c \leftarrow 1$, T_0, β
4: $\mathcal{A}_i \leftarrow \texttt{Random\_Selection}(\mathcal{V}_i, a)$
5: $\mathcal{P}_i \leftarrow \texttt{Pairwise\_Conn}(\mathcal{G} \setminus \mathcal{A}_i)$
6: **while** $c \leq$ iterations **do**
7: $\quad \mathcal{A}'_i \leftarrow \texttt{Perturb}_{\texttt{attack}}(\mathcal{A}_i, \mathcal{V}_i)$
8: $\quad \mathcal{P}'_i \leftarrow \texttt{Pairwise\_Conn}(\mathcal{G} \setminus \mathcal{A}'_i)$
9: $\quad p \leftarrow e^{-(\mathcal{P}'_i - \mathcal{P}_i)/T}$
10: $\quad$ **if** $(\mathcal{P}'_i < \mathcal{P}_i) \ \vee \ (\texttt{rand}(0,1) \leq p)$ **then**
11: $\quad\quad \mathcal{A}_i \leftarrow \mathcal{A}'_i$, $\mathcal{P}_i \leftarrow \mathcal{P}'_i$
12: $\quad$ **end if**
13: $\quad T \leftarrow T_0 \cdot e^{-\beta c}$
14: $\quad c \leftarrow c + 1$
15: **end while**
16: **return:** $\mathcal{A}_i$

If there are a total of d types of nodes, then to find the *best attack*, we apply the algorithm for each type, and then select the type i and the set of nodes $\mathcal{A}_i$ that result in the smallest pairwise connectivity.

In Algorithm 1, instead of random initialization of $\mathcal{A}_i$ (line 4), we can use a greedy solution obtained by Algorithm 2. This greedy solution is typically better than the random selection, thus providing a good starting point to the Algorithm 1. Along the lines in [27], We use the notion of *maximal independent set*, defined below, in our greedy approach.

Definition An independent set in a graph is a set of vertices, all of which are pairwise non-adjacent. If an independent set is such that it is not a subset of any other independent set, then it is called a *maximal independent set (MIS)*.

Note that if all nodes are of the same type, and the residual graph after an attack contains only such nodes that are in an independent set, then the pairwise connectivity of the residual graph is zero, and the attack is optimal [27]. Therefore, an attack consisting of nodes in the complement of MIS could be a good choice. Thus, the basic idea is to compute a complement of MIS, and then greedily select a nodes to compute an attack. Since in our attack model, non-trusted nodes of only one type i could be included in an attack, we first construct an alternate graph $\bar{\mathcal{G}}_i$ as follows: For an attack consisting of nodes of type i, obtain a subgraph induced by $\mathcal{V}_i$. We then add d more vertices to this subgraph, one for each of the remaining types and one for the trusted nodes. Let these new vertices be denoted by $\{v_j\} \cup \{t\}$, $j \neq i$. If $u \in \mathcal{V}_i$ is adjacent to some node of type $j \neq i$ in $\mathcal{G}$, then we add an edge between u and v_j in $\bar{\mathcal{G}}_i$. Similarly, if u is adjacent to some trusted node in $\mathcal{G}$, then we add an edge between u and t in $\bar{\mathcal{G}}_i$. An illustration of such a construction is given in Figure 2. Once $\bar{\mathcal{G}}_i$ is obtained, we compute a MIS $\mathcal{M}$, and then $\mathcal{V}_i \setminus \mathcal{M}$ (line 5), from which we construct an attack $\mathcal{A}_i$ consisting of a nodes (lines 6–13).



(a) $\mathcal{G}$ (b) $\bar{\mathcal{G}}_i$

Fig. 2: (a) A graph $\mathcal{G}$ with three node types and trusted nodes, in which we need to compute an attack $\mathcal{A}_1$ of nodes of type 1. (b) A construction of $\bar{\mathcal{G}}_i$ from $\mathcal{G}$.

For the defender, the objective is to select the trusted nodes as well as to assign a node type to each node. We again use a simulated annealing approach as outlined in Algorithm 3. Initially a set of t random nodes are selected as trusted, and each non-trusted node is assigned a type from the set $\mathcal{D}$ randomly. Subsequently, in each iteration, we search for a better $\mathcal{T}$ and Γ (type assignment to nodes) by perturbing the solution, as outlined in Algorithm 4. In the perturbation step, we replace a randomly selected trusted node in $\mathcal{T}$ by a randomly chosen node in the best attack $\mathcal{A}$ to obtain a new set of trusted nodes $\mathcal{T}'$. Similarly, we randomly change the type of an arbitrarily picked node in each $\mathcal{V}_i$ (lines 7–9 in Algorithm 4) to obtain a new type assignment Γ'. With this new $\mathcal{T}'$ and Γ', we compute a best attack $\mathcal{A}'$, and then the resulting pairwise connectivity $\mathcal{P}'$. If $\mathcal{T}'$ and Γ' is better than $\mathcal{T}$ and Γ in terms of improving the pairwise connectivity as a result of best attack, we select $\mathcal{T}'$ and Γ' as our new solution. As with the attacker's case, with small probability that decreases with the number of iterations, we select $\mathcal{T}'$ and Γ' as our new solution even if they result in a smaller pairwise connectivity as compared to $\mathcal{T}$ and Γ. Such sporadic replacements prevent the algorithm from getting stuck in a local minimum.

Algorithm 2 Attack Using Greedy Approach

1: **Input** $\mathcal{G}$, Γ, a
2: **Output** $\mathcal{A}_i$
3: Find a graph $\bar{\mathcal{G}}_i$
4: $\mathcal{M} \leftarrow \texttt{Max\_Ind\_Set}(\bar{\mathcal{G}}_i)$
5: $\mathcal{X} \leftarrow \mathcal{V}_i \setminus \mathcal{M}$, $\mathcal{A}_i \leftarrow \emptyset$
6: **if** $|\mathcal{X}| \leq a$ **then** $\mathcal{A}_i \leftarrow \mathcal{X}$
7: **else**
8: **while** $|A|_i \leq a$ **do**
9: $\alpha' \leftarrow \arg\min_{\alpha \in \mathcal{X}} \texttt{Pairwise\_Conn}(\mathcal{G} \setminus (\mathcal{A}_i \cup \{\alpha\}))$
10: $\mathcal{A}_i \leftarrow \mathcal{A}_i \cup \{\alpha'\}$
11: $\mathcal{X} \leftarrow \mathcal{X} \setminus \{\alpha'\}$
12: **end while**
13: **end if**
14: **return:** $\mathcal{A}_i$

Algorithm 3 Simulated Annealing Algorithm for the Defender

1: **Input** $\mathcal{G}$, $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_d\}$, a, t, iterations.
2: **Output** $\mathcal{T}$, Γ.
3: **Initialize:** $c \leftarrow 1$, T_0, β
4: $\mathcal{T} \leftarrow \texttt{Random\_Selection}(\mathcal{V}, t)$
5: Randomly select Γ (i.e., randomly assign a type from $\mathcal{D}$ to each $v \in \mathcal{V} \setminus \mathcal{T}$).
6: $\mathcal{A} \leftarrow \min_{i \in \{1,\cdots,d\}} \mathcal{A}_i = \texttt{Attack}(\mathcal{G}, \mathcal{V}_i, a)$
7: $\mathcal{P} \leftarrow \texttt{Pairwise\_Conn}(\mathcal{G} \setminus \mathcal{A})$
8: **while** $c \leq$ iterations **do**
9: $(\mathcal{T}', \Gamma') \leftarrow \texttt{Perturb}_{\texttt{defend}}(\mathcal{G}, \mathcal{T}, \Gamma, \mathcal{A}, \{\mathcal{A}_1, \cdots, \mathcal{A}_d\})$
10: $\mathcal{A}' \leftarrow \min_{i \in \{1,\cdots,d\}} \mathcal{A}'_i = \texttt{Attack}(\mathcal{G}, \mathcal{V}'_i, a)$
11: $\mathcal{P}' \leftarrow \texttt{Pairwise\_Conn}(\mathcal{G} \setminus \mathcal{A}')$
12: $p \leftarrow e^{-(\mathcal{P}-\mathcal{P}')/T}$
13: **if** $(\mathcal{P}' > \mathcal{P}) \ \vee \ (\texttt{rand}(0,1) \leq p)$ **then**
14: $\mathcal{T} \leftarrow \mathcal{T}'$, $\Gamma \leftarrow \Gamma'$, $\mathcal{P} \leftarrow \mathcal{P}'$
15: $\mathcal{A} \leftarrow \mathcal{A}'$, $\mathcal{A}_i \leftarrow \mathcal{A}'_i, \ \forall i$.
16: **end if**
17: $T \leftarrow T_0 \cdot e^{-\beta c}$
18: $c \leftarrow c + 1$
19: **end while**
20: **return:** $\mathcal{T}$, $\mathcal{D}$.

In the next section, we demonstrate that network's structural robustness, as measured by the pairwise connectivity, is significantly increased by having different types of nodes,

Algorithm 4 $\texttt{Perturb}_{\texttt{defend}}$

Input: $\mathcal{G}$, $\mathcal{T}$, Γ, $\mathcal{A}$, $\{\mathcal{A}_1, \cdots, \mathcal{A}_d\}$.
Output: $\mathcal{T}'$, Γ'.
Initialize: $\Gamma' \leftarrow \Gamma$
Randomly select $\alpha \in \mathcal{A}$, and $\tau \in \mathcal{T}$.
$\mathcal{T}' \leftarrow (\mathcal{T} \setminus \{\tau\}) \cup \{\alpha\}$.
Randomly assign a type from $\mathcal{D}$ to node τ.
for $i = 1 : d$ **do**
    Randomly select a node $v \in \mathcal{V}_i$, and randomly assign a type from $\mathcal{D} \setminus \{\mathcal{D}_i\}$ to v.
end for
return: $\mathcal{T}'$, Γ'.

some of which are also trusted.

VI. Numerical Evaluation

We illustrate our results on three benchmark networks from [26]. The network details are given below (for further details, see [26]),

- $[ER-235]$ The graph is generated using an Erdös-Rényi graph model, and has 235 nodes and 349 edges.
- $[ER-465]$ This is also an Erdös-Rényi graph with 465 nodes and 699 edges.
- $[BA-500]$ This graph is generated using a Barabási-Albert model using the preferential attachment mechanism, and has 500 nodes and 499 edges.

In Figure 3, for all three networks, we plot pairwise connectivity as a function of the number of attacked nodes computed using the Algorithm 1 (considering $T_0 = 0.01$ and $\beta = 30/\text{iterations}$). The details of plots are below.

- For each network, first we plot the pairwise connectivity for the base case, one in which there is no diversity with all nodes belonging to the same type, and there are no trusted nodes either.
- Then, we plot pairwise connectivity with no diversity $|\mathcal{D}| = 1$, but a subset of nodes being trusted (that is, $|\mathcal{T}| = 20, 40$ for ER-235; and $|\mathcal{T}| = 25, 50$ for both ER-465 and BA-500).
- We also plot the case in which there are no trusted nodes, but there are two types of non-trusted nodes ($|\mathcal{D}| = 2$).
- Finally, for each of the three networks, we plot the case in which both the diversity and trusted nodes are present (that is, $|\mathcal{T}| = 20$, $|\mathcal{D}| = 2$ for ER-235; and $|\mathcal{T}| = 25$, $|\mathcal{D}| = 2$ for both ER-465 and BA-500).

The selection of trusted nodes as well as nodes' types is obtained using the Algorithm 3 (considering $T_0 = 0.01$ and $\beta = 30/\text{iterations}$).

The plots in Figure 3 clearly indicate the significance of having diverse and trusted nodes to improve the pairwise connectivity of the residual graph after the attack. As the attack size increases, the pairwise connectivity drastically decreases if $\mathcal{T} = \emptyset$ and $|\mathcal{D}| = 1$. However, we see a significant improvement with a small fraction of trusted nodes and with two types of non-trusted nodes. In fact, we observe a much smaller rate of decrease of pairwise connectivity with an increase in the attack size if we have few trusted nodes and $|\mathcal{D}| = 2$.

In the case of both ER graphs, we observe that having only trusted nodes with no diversity is not the most effective strategy, even if we double the number of trusted nodes. On the other hand, incorporating only diversity $|\mathcal{D}| = 2$ with no trusted nodes is better than using trusted nodes alone. However, the most useful strategy is the combination of trusted nodes and diversity that significantly improves the pairwise connectivity. In the case of BA-500, we observe that using only diversity is almost inconsequential. In comparison, using only trusted nodes ($|\mathcal{T}| = 25$) gives much better results. In fact, combining diversity with 25 trusted nodes produces significant improvement in the pairwise connectivity. Interestingly, doubling the number of trusted nodes alone gives even better results.

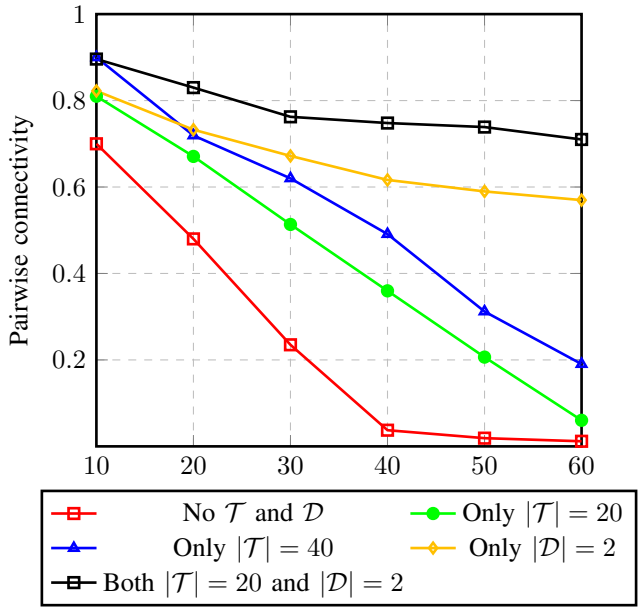
In Figure 4(a), we illustrate Algorithm 1 that computes the best attack consisting of nodes of a particular type. We consider the ER-235 network with an attack consisting of 50 nodes, $|\mathcal{T}| = 20$, $|\mathcal{D}| = 2$, and plot the pairwise connectivity as a function of the number of iterations. After about 1,000 iterations, the decrease in pairwise connectivity is very small suggesting the computation of a near-optimal attack. Similarly, in Figure 4(b), we illustrate Algorithm 3 that computes $\mathcal{T}$ and Γ for the same network. Again, we see a rapid increase in the pairwise connectivity initially and then the plot begins to flatten after about 2,000 iterations suggesting the computation of a near-optimal defender solution, that is $\mathcal{T}$ and Γ.

VII. Conclusions

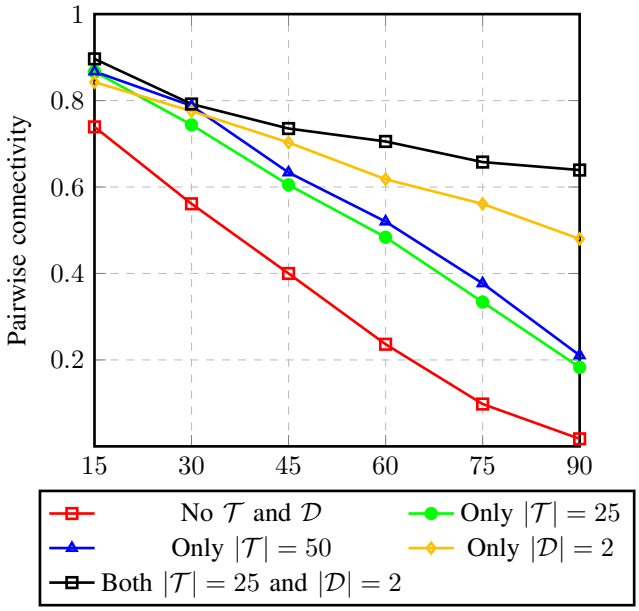
By having multiple variants of nodes (diversity) and a small number of hardened nodes that are insusceptible to failures (trustedness) in a network, we can significantly improve its structural robustness. In particular, damage caused by an optimal attack that removes a certain number of nodes from the network to minimize the pairwise connectivity of the residual network, is substantially reduced in the presence of diverse and trusted nodes. Thus, instead of solely relying on the typical redundancy approach, in which extra links are added between nodes, a combination of diversity and trustedness approaches can be employed effectively to increase networks' robustness against node removals, especially in the case of sparse networks. In this paper, we studied the specific measure of network robustness, namely pairwise connectivity, however, the notions of diverse and trusted nodes can be employed adequately to improve a broad spectrum of other network utility measures, which is also our future direction. Moreover, we would also like to develop more efficient algorithms to effectively distribute nodes of various types as well as trusted nodes within the network to maximize the effects of diversity and trustedness towards improving network's structural robustness.

References

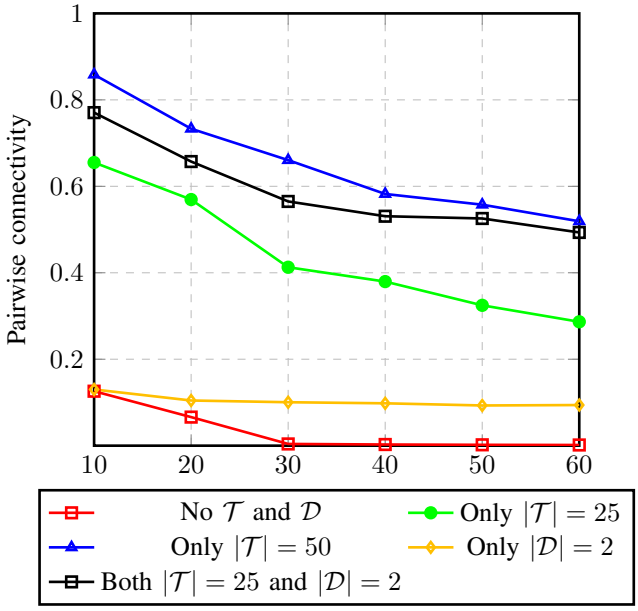
[1] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.



(a) ER-235



(b) ER-465



(c) BA-500

Fig. 3: Pairwise connectivity as a function of the number of attacked nodes for (a) ER-235, (b) ER-465, and (c) BA-500 networks.

[2] A. Mitra and S. Sundaram, "Secure distributed observers for a class of linear time invariant systems in the presence of byzantine adversaries," in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2709–2714.

[3] D. Bauer, H. Broersma, and E. Schmeichel, "Toughness in graphs – A survey," *Graphs and Combinatorics*, vol. 22, no. 1, pp. 1–35, 2006.

[4] M. Cozzens, D. Moazzami, and S. Stueckle, "The tenacity of a graph," *Graph theory, Combinatorics, and Algorithms*, vol. 1, p. 2, 1995.

[5] K. Edwards and G. Farr, "Fragmentability of graphs," *Journal of Combinatorial Theory, Series B*, vol. 82, no. 1, pp. 30–37, 2001.

[6] J. Wu, M. Barahona, Y.-J. Tan, and H.-Z. Deng, "Spectral measure of structural robustness in complex networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1244–1252, 2011.

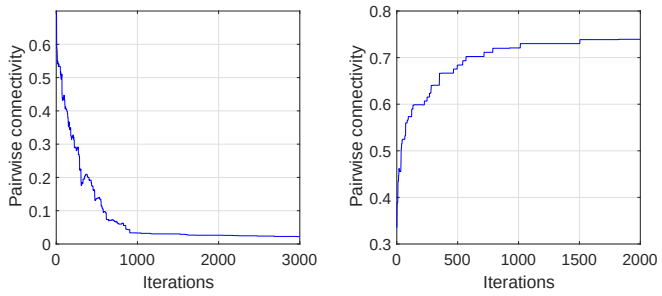
[7] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Y. Zomaya, "On the characterization of the structural robustness of data center networks," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 64–77, 2013.

[8] A. Newell, D. Obenshain, T. Tantillo, C. Nita-Rotaru, and Y. Amir, "Increasing network resiliency by optimally assigning diverse variants to routing nodes," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 602–614, 2015.

[9] M. Ventresca and D. Aleman, "Efficiently identifying critical nodes in large complex networks," *Computational Social Networks*, vol. 2, no. 1, p. 6, 2015.

[10] T. N. Dinh, Y. Xuan, M. T. Thai, P. M. Pardalos, and T. Znati, "On new approaches of assessing network vulnerability: hardness and approximation," *IEEE/ACM Transactions on Networking (ToN)*, vol. 20, no. 2, pp. 609–619, 2012.

[11] M. F. Habib, M. Tornatore, F. Dikbiyik, and B. Mukherjee, "Disaster survivability in optical communication networks," *Computer Communications*, vol. 36, no. 6, pp. 630–644, 2013.



(a) (b)

Fig. 4: (a) Pairwise connectivity as a function of number of iterations in Algorithm 1. (b) Pairwise connectivity as a function of iterations in Algorithm 3. In both plots, we consider ER-235 network, an attack of 50 nodes, $|\mathcal{T}| = 20$ and $|\mathcal{D}| = 2$.

[12] T. N. Dinh, Y. Xuan, M. T. Thai, E. Park, and T. Znati, "On approximation of new optimization methods for assessing network vulnerability." in *INFOCOM*, 2010, pp. 1–9.

[13] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.

[14] G. N. Frederickson and J. Ja'Ja', "Approximation algorithms for several graph augmentation problems," *SIAM Journal on Computing*, vol. 10, no. 2, pp. 270–283, 1981.

[15] G. Kortsarz and Z. Nutov, "Tight approximation algorithm for connectivity augmentation problems," *Journal of Computer and System Sciences*, vol. 74, no. 5, pp. 662–670, 2008.

[16] L. A. Végh, "Augmenting undirected node-connectivity by one," *SIAM Journal on Discrete Mathematics*, vol. 25, no. 2, pp. 695–718, 2011.

[17] Z. Nutov, "Approximating node-connectivity augmentation problems," *Algorithmica*, vol. 63, no. 1-2, pp. 398–410, 2012.

[18] S. Forrest, A. Somayaji, and D. H. Ackley, "Building diverse computer systems," in *The Sixth Workshop on Hot Topics in Operating Systems*, 1997, pp. 67–72.

[19] J. Xu, Z. Kalbarczyk, and R. K. Iyer, "Transparent runtime randomization for security," in *Proceedings of the 22nd International Symposium on Reliable Distributed Systems (SRDS)*, 2003, pp. 260–269.

[20] A. J. O'Donnell and H. Sethu, "On achieving software diversity for improved network security using distributed coloring algorithms," in *Proceedings of the 11th ACM conference on Computer and Communications Security (CCS)*, 2004, pp. 121–131.

[21] A. Alarifi and W. Du, "Diversify sensor nodes to improve resilience against node compromise," in *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2006, pp. 101–112.

[22] Y. Yang, S. Zhu, and G. Cao, "Improving sensor network immunity under worm attacks: a software diversity approach," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008, pp. 149–158.

[23] W. Abbas, A. Laszka, and X. Koutsoukos, "Improving network connectivity and robustness using trusted nodes with application to resilient consensus," *IEEE Transactions on Control of Network Systems*, 2017.

[24] W. Abbas, A. Laszka, Y. Vorobeychik, and X. Koutsoukos, "Improving network connectivity using trusted nodes and edges," in *American Control Conference (ACC)*. IEEE, 2017, pp. 328–333.

[25] A. Mitra, W. Abbas, and S. Sundaram, "On the impact of trusted nodes in resilient distributed state estimation of LTI systems," in *IEEE Conference on Decision and Control (CDC)*, 2018.

[26] M. Ventresca, "Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem," *Computers & Operations Research*, vol. 39, no. 11, pp. 2763–2775, 2012.

[27] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Computers & Operations Research*, vol. 36, no. 7, pp. 2193–2200, 2009.

[28] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia, "Local search metaheuristics for the critical node problem," *Networks*, vol. 67, no. 3, pp. 209–221, 2016.

[29] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Computer Science Review*, vol. 28, pp. 92–117, 2018.