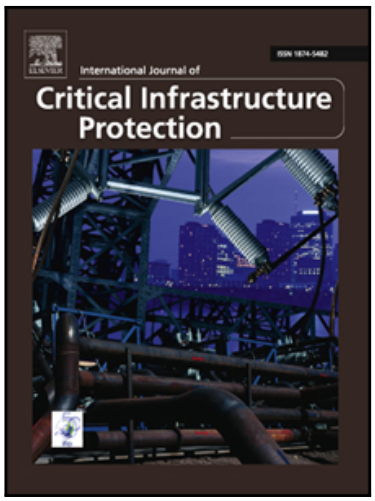


Finding critical nodes in infrastructure networks

Luca Faramondi, Roberto Setola, Stefano Panzieri,
Federica Pascucci, Gabriele Oliva

PII: S1874-5482(17)30091-4
DOI: 10.1016/j.ijcip.2017.11.004
Reference: IJCIP 234

To appear in: *International Journal of Critical Infrastructure Protection*

Received date: 18 May 2017
Revised date: 23 October 2017
Accepted date: 15 November 2017

Please cite this article as: Luca Faramondi, Roberto Setola, Stefano Panzieri, Federica Pascucci, Gabriele Oliva, Finding critical nodes in infrastructure networks, *International Journal of Critical Infrastructure Protection* (2017), doi: 10.1016/j.ijcip.2017.11.004

Finding critical nodes in infrastructure networks

Luca Faramondi[a,b,1], Roberto Setola[a,b], Stefano Panzieri[c], Federica Pascucci[c], Gabriele Oliva[a,b]

[a] *Automatic Control Unit, Department of Engineering, Università Campus Bio-Medico di Roma, via Álvaro del Portillo 21, 00128 Rome, Italy*
[b] *Consorzio Nazionale Interuniversitario per i Trasporti e la Logistica (NITEL), via Spalato 11, 00198 Rome Italy*
[c] *Department of Engineering, University of Rome Tre, via della Vasca Navale 79, 00146 Rome, Italy*

Abstract

It is well known that profiling attacker behavior is an effective way to obtain insights into network attacks and to identify the systems and components that must be protected. This paper presents a novel integer linear programming formulation that models the strategy of an attacker who targets a set of nodes with the goal of compromising or destroying them. The attacker model considers the infliction of the greatest possible damage with minimal attacker effort. Specifically, it is assumed that the attacker is guided by three conflicting objectives: (i) maximization of the number of disconnected components; (ii) minimization of the size of the largest connected component; and (iii) minimization of the attack cost. Compared with other research in the area, the proposed formulation is much more descriptive but has less complexity; thus, it is very useful for predicting attacks and identifying the entities that must be protected. Since exact solutions of the formulation are computationally expensive for large problems, a heuristic algorithm is presented to obtain approximate solutions. Simulation results using a U.S. airport network dataset demonstrate the effectiveness and utility of the proposed approach.

Keywords

Critical Infrastructure Networks; Critical Nodes; Attacker Perspective; Attacker Profiling

[1] Corresponding author: Luca Faramondi (l.faramondi@unicampus.it)

Manuscript No.: IJCIP-2017-56
Submitted: May 18, 2017; Revision 1: July 18, 2017; Revision 2: October 23, 2017
Accepted: November 15, 2017

1. Introduction

The seminal research of Albert et al. [1] and Holme et al. [13] in the early 2000s have demonstrated that attacks that take into account the topological structures of networks (e.g., telecommunications networks and road infrastructures) can have dramatic consequences. An attacker who knows the network topology can select the target sites more effectively, increasing the damage (e.g., by disconnecting large portions of the network) while keeping the cost of the attack at a minimum. Interested readers are referred to [5, 14, 17, 20, 26] for recent research on this topic.

This research assumes that the attacker objectives are to cause the maximum damage with the minimum attack cost. Such attacker profiling is an effective technique for identifying and managing critical infrastructure vulnerabilities – it is the first step in implementing appropriate risk mitigation strategies. Indeed, this step is extremely important – as highlighted, for instance, by the European Commission Directive on the Security of Network and Information Systems [11], which requires critical infrastructure operators, and specifically information technology providers, to take adequate measures to manage risk, report security incidents to national authorities and provide early warnings of threats.

The main contribution of this paper is a methodology that identifies the critical nodes to be protected based on the conflicting attacker objectives of maximizing the damage while minimizing the attack cost. The methodology characterizes the behavior of an attacker who targets some of the nodes in a network by corrupting or disrupting them. In the proposed formulation, the profiled attacker seeks to have a large impact with limited resources. To achieve this, the attacker attempts to identify nodes that, if removed, divide the network into many small partitions, allowing each node to communicate only with a small subset of nodes. The problem is framed as an integer linear programming problem by introducing suitable constraints that will be discussed later. When solving the problem, it is not necessary to specify a fixed number of partitions; this is because the maximization of the number

of disconnected components becomes an additional attacker objective that is considered along with the minimization of the size of the largest component and the minimization of the attack cost. Indeed, the proposed formulation is an improvement over previous approaches. By removing the constraints on fixed parameters such as the number of partitions [12], largest partition size [3] and attack cost [2], it is possible to better reproduce attacker behavior by shifting the focus to attacker preferences without making assumptions about the features of the final solution.

Like other related techniques (see, e.g., [21]), the proposed approach requires $O(n^2)$ Boolean decision variables, where n is the number of nodes in the network of interest. Since an exact solution is computationally expensive for a large problem, a heuristic algorithm is presented that finds a sub-optimal solution in a cost-effective manner. Simulation results using a U.S. airport network dataset demonstrate the effectiveness and utility of the approach.

2. Related work

Several techniques based on operations research and graph theory have been applied to critical infrastructure protection problems. The techniques involve estimating and analyzing resilience in infrastructure networks based on (constrained) optimization problems [2, 3, 9, 10, 12, 15, 21, 24], leveraging bi-level optimization frameworks [6, 7] and employing network spectral analysis [16, 18, 22, 27]. Some techniques model infrastructures in terms of graphs and evaluate their robustness by identifying the critical nodes. Others consider critical links [8, 21, 25] that when removed degrade a connection-related index such as the average inverse geodesic length (i.e., the sum of the inverses of the shortest paths between pairs of nodes) or the total pairwise connectivity [23].

Due to the computational complexity associated with such techniques, critical nodes are identified using graph spectral analysis (see, e.g., [20] where percolation theory is used to assess network robustness). In these instances, metrics such as the node degree or eigenvalues of the adjacency matrix are used to evaluate network robustness. Lu and Li [18] have estimated network vulnerabilities based on the structural controllability of a network after iteratively removing nodes based on node-degree order, eigenvector centrality and betweenness. Such metrics are commonly employed in critical infrastructure

protection problems (see, e.g., [22, 27] and the references they contain) in order to assess node criticality and to articulate risk mitigation strategies.

Many of these approaches have been developed for and applied to electrical networks. The approaches focus on the ability of a network to remain connected without considering specific source-destination links or specific paths. This aspect is incorporated when studying critical node disruptor problems [2, 10, 21]. In this type of problem, an attacker targets certain nodes in a network to minimize the total pairwise connectivity [23], which is the number of node pairs that are connected by a path after the attacked nodes are removed. However, the approach assumes that the attacker has either *a priori* knowledge of the maximum number of nodes k that must be disconnected or the ability to disconnect up to a fixed number of nodes k; an incorrect choice of k may result in an infeasible or inefficient solution. These approaches are also useful for estimating the effectiveness of attacks on communications and transportation networks in which operational channels/links must exist between sources and destinations.

Pullan [19] has focused on the dual problem of cardinality-constrained critical node detection where the maximum allowable size of connected graph components is specified and the goal is to minimize the number of attacked nodes that satisfies this constraint. Ventresca et al. [24] believe that the critical node disruptor problem, which is intrinsically multi-objective, can be better phrased in terms of minimizing the pairwise connectivity and the variance in the cardinality of the connected components (i.e., "islands" obtained after removing nodes). However, this problem specification has the same drawbacks as the standard critical node disruptor problem specification. The approach of Lalou et al. [15] is structured along similar lines; specifically, the size of each connected component is constrained to be less than a certain value. In contrast, the approach proposed in this paper is based on the research by Faramondi et al. [12], which assumes that an attacker is not constrained to target a fixed number of nodes. Instead, the attacker seeks to divide a network into a fixed number of components in the face of two conflicting objectives: (i) minimizing the number of attacked nodes; and (ii) minimizing the size of the largest component.

3. Preliminaries

In the following presentation, $|X|$ denotes the cardinality of a set X; vectors are represented using boldface letters and $\mathbf{k}_m$ denotes a vector in $\mathbb{R}^m$

whose components are all equal to k. The term $0_{n,m}$ denotes an $n \times m$ matrix whose entries are all 0. The $n \times n$ identity matrix is denoted by I_n.

Let A be an $n \times m$ matrix A and B be a $p \times q$ matrix. The Kronecker product of A and B is the $np \times mq$ matrix given by:

$$A \otimes B = \begin{bmatrix} A_{11}B & \dots & A_{1m}B \\ \vdots & \ddots & \vdots \\ A_{n1}B & \dots & A_{nm}B \end{bmatrix}$$

Given a matrix Q, Q_+ and Q_- denote its non-negative and non-positive parts, respectively. These two matrices have the same dimensions as Q, but contain the non-negative and non-positive entries of Q, respectively, while the other entries are zeros. Therefore, it follows that $Q = Q_+ + Q_-$.

Let $G = \{V, E\}$ be a graph with n nodes $V = \{v_1, v_2, \dots, v_n\}$ and e edges $E \subseteq V \times V$, where $(v_i, v_j) \in E$ expresses the existence of a relation between nodes v_i and v_j. A partition $V_i \subseteq V$ is a subset of the nodes in V.

A graph is undirected if $(v_i, v_j) \in E$ whenever $(v_j, v_i) \in E$; otherwise, the graph is directed. The remainder of this paper only considers undirected graphs.

A path in a graph $G = \{V, E\}$ starting at node $v_i \in V$ and ending at node $v_j \in V$ is a subset of the links in E that connect v_i and v_j respecting the edge orientation and without creating loops. The length of a path is the number of links in the path; the minimum path is the path of minimum cardinality.

An undirected graph is connected if, for every pair of nodes v_i, v_j, there exists a path in the graph that connects the pair of nodes. The neighborhood $\mathcal{N}(v_i)$ of a node $v_i \in V$ is the set of vertices connected to v_i by an edge in E.

The adjacency matrix of a graph G is an $n \times n$ matrix A such that $A_{ij} = 1$ if $(v_j, v_i) \in E$ and $A_{ij} = 0$ otherwise.

The incidence matrix of a graph G is an $e \times n$ matrix M such that each row represents a link and, for a link $x = (v_i, v_j) \in E$ and a node y, the following equation holds:

$$M_{xy} = \begin{cases} 1 & \text{if } y = i \\ -1 & \text{if } y = j \\ 0 & \text{otherwise} \end{cases}$$

In an undirected graph with n nodes, at most $\frac{n(n-1)}{2}$ distinct pairs of nodes are connected via a path. The pairwise connectivity (PWC) [23] of a

graph G is defined as:

$$PWC(G) = \frac{1}{n(n-1)} \sum_{v_i, v_j \in V, v_i \neq v_j} p(v_i, v_j), \tag{1}$$

where $p(v_i, v_j) = 1$ if the pair (v_i, v_j) is connected by a path in G; otherwise, $p(v_i, v_j) = 0$. The pairwise connectivity is a measure of connectivity that considers the existence of a path between any pair of nodes. The maximum $PWC(G) = 1$ occurs when the graph is connected. In the following, the pairwise connectivity PWC is expressed as a percentage (i.e., $PWC = 100 \times PWC(G)$).

In an undirected graph, the pairwise connectivity is a function of the size $|V_i|$ of each connected component $G_i = \{V_i, E_i\}$ of the graph after the attacked nodes have been removed along with their incident edges. In fact, for each connected component G_i of graph G, there are exactly $|V_i|(|V_i| - 1)/2$ distinct pairs of connected nodes. If graph G contains m connected components, then Equation (1) can be rewritten as:

$$PWC(G) = \frac{1}{n(n-1)} \sum_{i=1}^{m} |V_i|(|V_i| - 1) \tag{2}$$

(a) $PWC = 26.6\%$. (b) $PWC = 40\%$.

Figure 1: Node deletion in central and peripheral areas.

Figure 1 shows that node deletion can have remarkably different effects on pairwise connectivity. In Figure 1(a), when the central node is removed, the graph is disconnected into two components and only four pairs of nodes are connected by a path; thus, $PWC = 26.6\%$. In Figure 1(b), the graph is decomposed into two components again, but six pairs of nodes are connected by a path and $PWC = 40\%$. The figures suggest that disconnected graphs whose connected components have balanced sizes have lower pairwise connectivity values compared with disconnected graphs whose connected components do not have balanced sizes.

4. Optimization problem formulation

The optimization problem objective is to find the minimum number of nodes – called critical nodes – whose removal causes a performance degra-

dation in the network in terms of the pairwise connectivity PWC. The multi-objective optimization problem has two conflicting objectives: (i) minimization of network connectivity; and (ii) minimization of attack cost. When framing the problem as an integer linear programming formulation, it is assumed that the attacker is interested in finding the optimal solution that:

- Maximizes the number of network partitions.
- Minimizes the maximum network partition cardinality.
- Minimizes the attack cost.

An examination of Equation (2) and Figure 1 reveals that the maximization of the number of network partitions and the simultaneous minimization of the maximum network partition cardinality correspond to the minimization of the pairwise connectivity.

Let $G = \{V, E\}$ be a undirected connected graph and assume that the attacker selects some nodes $v_j \in V_C \subseteq V$. As a result of the attack, at most $n-1$ connected components can be obtained. For example, in a star graph, if the central node is attacked, the network is disconnected into $n-1$ components (i.e., each is an isolated node).

Consider a set of $n-1$ pairwise disjoint partitions $V_1, \ldots, V_{n-1}$ such that $V = V_C \cup_{i=1,\ldots,n-1} V_i$. Furthermore, consider the Boolean variables $x_j^{(i)}$ ($j = 1, \ldots, n$; $i = 1, \ldots, n-1$) such that $x_j^{(i)} = 1$ when node v_j is assigned to the partition V_i. Let c_j be a Boolean variable such that $c_j = 1$ if $v_j \in V_C$; otherwise, $c_j = 0$.

In order to model real scenarios, a vector $\mathbf{k}$ with n entries is introduced, where $k_i \in [0, 1]$ and n is the number of network nodes. Each entry k_i expresses the cost associated with removing the i^{th} node.

The partitions $V_1, \ldots, V_{n-1}$ (clarified later) reflect how the nodes in the graph are separated after an attack. Specifically, in order to characterize the attack, these nodes, which are labeled as belonging to different partitions, should not be connected by a path after the attacked nodes in V_C have been removed. However, there is no guarantee that a partition contains a single connected component (i.e., each partition might be further decomposed into connected components).

In the proposed formulation, some partitions may be empty. In fact, the maximization of the number of non-empty partitions is an attacker objective.

In order to model the ability of an attacker to select an arbitrary number of non-empty partitions, it is necessary to associate a Boolean variable t_i with each partition V_i. If V_i is empty, then $t_i = 0$; otherwise, $t_i = 1$.

Finally, the three optimization objectives can be defined as follows:

- *Objective 1:* Maximize the number of disjoint and non-empty partitions:

$$\max \sum_{t=1}^{n-1} t_i \tag{3}$$

- *Objective 2:* Minimize the maximum partition cardinality:

$$\min \max_{V_1,\ldots,V_{n-1}} |V_i| \tag{4}$$

- *Objective 3:* Minimize the attack cost:

$$\min \sum_{i=1}^{n} k_i c_i \tag{5}$$

4.1. Problem constraints

Certain constraints are introduced to obtain an integer linear programming formulation.

The first set of constraints requires each node to be assigned to a single set in $V_C, V_1, \ldots, V_{n-1}$:

$$\mathbf{c} + \sum_{i=1}^{n-1} \mathbf{x}^{(i)} = \mathbf{1}_n \tag{6}$$

where $\mathbf{c}$ and $\mathbf{x}^{(i)}$ are the stack vectors of the variables c_i and $x_j^{(i)}$, respectively.

In order to select decision variables that correspond to an attack that successfully divides the network into connected components, the nodes assigned to V_i must not be directly connected to the nodes in V_j for all $i, j = 1, \ldots, n-1; i \neq j$. In other words, the second set of constraints (which will be formally specifed later) requires the following condition to hold:

$$(v_a, v_b) \notin E \quad \text{where } v_a \in V_i;\ v_b \in V_j;\ \forall i, j = 1, \ldots, n-1 \tag{7}$$

Note that every pair of partitions V_i and V_j (not considering V_C) and every pair of nodes v_a and v_b must satisfy Equation (7). This condition can be expressed as:

$$A_{ab}\left(x_a^{(i)} + x_b^{(j)}\right) \leq 2 - \epsilon \tag{8}$$

where $0 < \epsilon < 1$ is a coefficient that is required to avoid the use of strict inequalities. The condition can be trivially verified when the coefficient A_{ab} of the adjacency matrix is zero (i.e., when $(v_a, v_b) \notin E$). Conversely, in the case of $(v_a, v_b) \in E$, the condition is violated when $v_a \in V_i$ and $v_b \in V_j$.

Let M be the incidence matrix of graph G and let M_+ and M_- be its non-negative and non-positive parts. The second set of constraints – which are called separation constraints – can be expressed in a compact form for a pair of sets V_i and V_j and for all the edges as follows:

$$\begin{aligned} M_+\mathbf{x}^{(i)} - M_-\mathbf{x}^{(j)} &\leq (2-\epsilon)\mathbf{1}_e \\ M_+\mathbf{x}^{(j)} - M_-\mathbf{x}^{(i)} &\leq (2-\epsilon)\mathbf{1}_e \end{aligned} \tag{9}$$

Note that two specular constraints are specified for each edge. Indeed, it is necessary to take into account undirected graphs and explicitly handle (v_i, v_j) and (v_j, v_i) for every pair of nodes that are connected by a link.

In order to maximize the number of non-empty partitions, it is necessary to introduce another set of constraints. This set of constraints describes the relation between the variables $x_j^{(i)}$ and t_i:

$$x_1^{(i)} + \ldots + x_n^{(i)} \geq t_i \quad i = 1 \ldots n-1 \tag{10}$$

According to these constraints, a partition is non-empty if it has at least one node assigned to itself.

The problem is further simplified by eliminating the min and max operators in Equation (4) and introducing a new free variable $q \in \mathbb{N}$ and the additional set of constraints (11):

$$\mathbf{1}_n^T\mathbf{x}^{(i)} \leq q \quad \forall i = 1, \ldots, n-1 \tag{11}$$

so that Equation (4) can be replaced as:

$$\min_{q \in \mathbb{N}} q$$

As a result, the set of constraints in Equation (11) requires:

$$q \geq \max_{i=1,\ldots,n-1}\left(\mathbf{1}_n^T\mathbf{x}^{(i)}\right) \tag{12}$$

Since q is minimized, the relation in Equation (12) is always satisfied as an equality.

The last set of constraints guarantees the presence of at least one critical node:

$$\sum_{i=1}^{n-1} \mathbf{1}_n^T \mathbf{x}^{(i)} \leq n-1 \tag{13}$$

In fact, this inequality is satisfied if at least one node is not assigned to any of the $n-1$ partitions $V_1, \ldots, V_{n-1}$. As a result of the constraints in Equation (6), the "missing" node belongs to the critical set.

4.2. Objective function

The objective function comprises the three objectives. A convex combination of the three objectives is created by introducing weights α_1, α_2 and α_3 such that:

$$\alpha_i \in [0,1];\ i \in \{1,2,3\};\ \sum_{i=1}^{3} \alpha_i = 1 \tag{14}$$

This definition permits the expression of a large set of attacker behaviors using different weights for the sub-objectives specified by Equations (3), (4) and (5).

Thus, the overall objective function is given by:

$$\min \Big\{ \underbrace{\frac{n}{\sum_{i=1}^{n} k_i} \alpha_1 \mathbf{k}^T \mathbf{c}}_{\text{Attack Cost}} + \underbrace{\alpha_2 q - \alpha_3 \sum_{i=1}^{n-1} t_i}_{\widehat{PWC}} \Big\} \tag{15}$$

The first term in Equation (15) expresses the attack cost, which depends on the number of critical nodes assigned in $\mathbf{c}$ and the associated costs k_i. In addition to employing the weight α_1 that expresses the attacker's preference, the attack cost is normalized to avoid three unbalanced sub-objectives in Equation (15). The other two terms in Equation (15) attempt to minimize an approximation of the pairwise connectivity denoted as $\widehat{PWC}$, which is considerably easier to compute than the standard pairwise connectivity PWC. Details about the relationship between the proposed approximation $\widehat{PWC}$ and the standard pairwise connectivity PWC are discussed in Section 6.

The proposed approach replaces the standard pairwise connectivity with the following approximation:

$$\widehat{PWC} = \gamma q - (1-\gamma)\sum_{i=1}^{n-1} t_i \tag{16}$$

such that $\gamma \in [0,1]$.

Next, a compact form that synthesizes the proposed integer linear programming formulation is presented. To this end, let

$$\mathbf{x} = \begin{bmatrix}(\mathbf{x}^{(1)})^T & \cdots & (\mathbf{x}^{(n-1)})^T\end{bmatrix}^T \quad \mathbf{t} = \begin{bmatrix}t_1 & \cdots & t_{n-1}\end{bmatrix}^T$$

The vector of independent variables is given by:

$$\mathbf{y} = \begin{bmatrix}\mathbf{c}^T, \mathbf{x}^T, q, \mathbf{t}^T\end{bmatrix}^T \tag{17}$$

In order to express the optimization problem in the standard integer linear programming form, the constraints in Equation (6) and (9) must be modified. Specifically, since the constraints in Equation (6) are in equality form, they must be expressed in terms of two inequality constraints:

$$\begin{array}{c} -\mathbf{c} - \sum_{i=1}^{n-1} \mathbf{x}^{(i)} \leq -\mathbf{1}_n \\ \mathbf{c} + \sum_{i=1}^{n-1} \mathbf{x}^{(i)} \leq \mathbf{1}_n \end{array} \tag{18}$$

With regard to the constraints in Equation (9), consider the matrix given by:

$$\mathcal{D} = \begin{bmatrix} M_+^{(n-1)} \otimes M_+ + M_-^{(n-1)} \otimes M_- \\ -M_+^{(n-1)} \otimes M_- - M_-^{(n-1)} \otimes M_+ \end{bmatrix} \tag{19}$$

where $M^{(n-1)}$ is the incidence matrix of a complete graph with $n-1$ nodes.

Note that $\mathcal{D}$ has $\xi = (n-1)(n-2)e$ rows. In fact, $M_+^{(n-1)}$ represents a complete graph with $n-1$ nodes and $\frac{1}{2}(n-1)(n-2)$ edges (hence, it has $\frac{1}{2}(n-1)(n+2)$ rows) while M_+ has e rows. As a consequence, the Kronecker product $M_+^{(n-1)} \otimes M_+$ has $\frac{1}{2}\xi$ rows.

Using $\mathcal{D}$, the constraints in Equation (9) can be expressed in a compact form as:

$$\mathcal{D}\mathbf{x} \leq (2-\epsilon)\mathbf{1}_\xi$$

At this point, each set of constraints is expressed in standard form. In the following, note that $\mathbf{y} \in \{0,1\}^{n+n^2} \cup \mathbb{N}$ means that the only natural variable in $\mathbf{y}$ is q and all the other entries in $\mathbf{y}$ are Boolean.

The resulting optimization problem is given by:

$$\min_{\mathbf{y}} \mathbf{r}^T\mathbf{y} \quad \text{subject to} \quad \begin{cases} \mathcal{A}\mathbf{y} \leq \mathcal{B} \\ \mathbf{y} \in \{0,1\}^{n+n^2} \cup \mathbb{N} \end{cases} \tag{20}$$

where the constraints are collected in the matrix $\mathcal{A}$ and vector $\mathcal{B}$:

$$\mathcal{A} = \begin{bmatrix} -I_n & -\mathbf{1}_{n-1}^T \otimes I_n & \mathbf{0}_n & 0_{n,n-1} \\ I_n & \mathbf{1}_{n-1}^T \otimes I_n & \mathbf{0}_n & 0_{n,n-1} \\ 0_{\xi,n} & \mathcal{D} & \mathbf{0}_\xi & 0_{\xi,n-1} \\ 0_{n-1,n} & -I_{n-1} \otimes \mathbf{1}_n^T & \mathbf{0}_{n-1} & I_{n-1,n-1} \\ 0_{n-1,n} & I_{n-1} \otimes \mathbf{1}_n^T & -\mathbf{1}_{n-1} & 0_{n-1,n-1} \\ 0_{1,n} & \mathbf{1}_{n-1}^T \otimes \mathbf{1}_n^T & 0 & \mathbf{0}_{n-1}^T \end{bmatrix} \mathcal{B} = \begin{bmatrix} -\mathbf{1}_n \\ \mathbf{1}_n \\ (2-\epsilon)\mathbf{1}_\xi \\ -\mathbf{1}_{n-1} \\ \mathbf{0}_{n-1} \\ n-1 \end{bmatrix}$$

$$\mathbf{r}^T = \begin{bmatrix} \frac{\alpha_1 n \mathbf{k}_n^T}{\mathbf{k}_n^T \mathbf{1_n}} & \mathbf{0}_{n(n-1)}^T & \alpha_2 & -\alpha_3 \mathbf{1}_{n-1}^T \end{bmatrix}$$

To clarify, the constraints in Equation (6) are collected in the first two rows of $\mathcal{A}$ and $\mathcal{B}$. The constraints in Equation (9) are expressed in the third rows of $\mathcal{A}$ and $\mathcal{B}$ with reference to the matrix $\mathcal{D}$ defined by Equation (19). Finally, the constraints in Equations (10), (11) and (13) are represented by the last three rows of $\mathcal{A}$ and $\mathcal{B}$, respectively. The components of vector $\mathbf{r}$ represent the costs by which the variables are weighted – the terms $\frac{\alpha_1 n \mathbf{k}_n^T}{\mathbf{k}_n^T \mathbf{1_n}}$, α_2 and $-\alpha_3 \mathbf{1}_{n-1}^T$ characterize the three sub-objectives.

Remark 1. It can be shown that the number of rows in matrix $\mathcal{A}$ is given by:

$$r_{\mathcal{A}} = 4n + (n-1)(n-2)e - 1$$

where n is the number of nodes and e is the number of edges in graph G. Therefore, the integer linear programming formulation specified by Equation (20) has $O(n^2 e)$ constraints.

As a consequence of this remark, sparse graphs with $e \ll n^2$ must satisfy a reduced number of constraints. On the other hand, dense graphs with $e \approx n^2$ must satisfy $O(n^4)$ constraints.

5. Heuristic algorithm

Due to the computational effort involved, it is infeasible to solve the integer linear programming problem exactly for a large network. As a conse-

quence, this section describes a heuristic approach that provides an approximate solution in reasonable time by sampling a large number of solutions and selecting the minimum cost solution from among the sampled solutions.

Since the integer linear programming problem has $O(n^2)$ Boolean variables with complex relations that must be verified (e.g., separation constraints), it is difficult to apply a brute-force Monte Carlo approach. In fact, there is a risk that a large fraction of the solutions generated by such an approach would be infeasible.

Since q and $\mathbf{t}$ depend on the node partitioning process, it is possible to easily find admissible choices for q and $\mathbf{t}$ given an admissible choice for the entries in $\mathbf{x}$. This intuition has led to the specification of an algorithm that generates feasible solutions (Algorithm 1).

(a) Feasible solution 1. (b) Feasible solution 2.

Figure 2: Heuristic assignment criterion adopted by Algorithm 1.

The feasible solution generation (FSG) algorithm assigns each node to a partition by considering its neighborhood. If all the already-assigned neighbors of a node v_i belong to the same partition, then a feasible solution is to assign v_i to the same partition (Figure 2(a)). However, if a node v_i has neighbors assigned to different partitions, then the only feasible choice is to set v_i as a critical node (C) to preserve the absence of links between the partitions (Figure 2(b)).

Algorithm 1 assumes that only $m \leq n-1$ partitions can be non-empty and it evaluate each node in V in random order. Specifically, each node v_i is assigned to a partition (or to the set of critical nodes) as described below.

If no neighbor of v_i is assigned to a partition (lines 15-26), then two possible sub-cases exist. In the first sub-case, if there is an empty partition left (lines 16-21), then v_i is randomly assigned to one of them (h) and the number of non-empty partitions ϕ is incremented by one. Additionally, for each neighbor v_j in $\mathcal{N}_i$, the list of assigned neighbors $\mathcal{M}_j$ is updated. In the second sub-case (lines 23-24), the node is reconsidered by inserting it back in the set of nodes that have not been considered. This procedure is performed a maximum of $\chi_{\max}$ times.

On the other hand, if at least one neighbor of v_i belongs to a partition (lines 27-34), then there are two possible sub-cases. In the first sub-case (lines 28-31), if all the already-assigned neighbors of v_i belong to the same

partition h, then v_i is assigned to the same partition h. Additionally, for each neighbor v_j of v_i, the list of assigned neighbors $\mathcal{M}_j$ is updated. In the second sub-case (lines 32-33), if none of the above cases is verified, then v_i has neighbors assigned to different partitions and v_i is labeled as critical.

The procedure ends when all the nodes have been assigned or when the maximum number of reconsiderations χ_{max} have been performed. In the latter case, note that not every assigned node is labeled as critical. Moreover, nodes are actively labeled as critical only at the end of the main cycle, using the constraints in Equation (6) after all the $x_j^{(i)}$ have been specified. Note also that, in order to eliminate feasible solutions with large numbers of critical nodes, a suitable choice for the parameter χ_{max} is at least one order of magnitude greater than the number of nodes n.

Algorithm 1 concludes by computing the $\mathbf{y}$ entries and q based on the partition assignments. In addition to ensuring feasible solutions, the algorithm guarantees the internal connectivity of each partition. Note that, in the problem formulation, there is no guarantee that the partitions are internally connected.

Algorithm 2 performs heuristic network vulnerability detection on the large number of solutions n_a provided by the feasible solution generation algorithm (Algorithm 1). It is necessary to show that Algorithm 2 always provides a feasible solution.

Remark 2. Algorithm 2 always provides a feasible solution. In fact, assigning a node to the set of critical nodes when its neighbors belong to more than one partition ensures that the separation constraints are enforced.

Algorithm 2 performs heuristic network vulnerability detection using a random integer number of allowed partitions m_{tmp}. Note that, in some cases (e.g., for large networks), the optimal solution is unlikely to contain a number of partitions (i.e., $O(n)$). Therefore, a possible choice is to arbitrarily fix the maximum number of partitions $m_{\max} \ll n$ in Algorithm 2. The algorithm selects a value $m_{tmp} \in \{2, \ldots, m_{\max}\}$ at each round with probability:

$$Pr(m_{tmp} = q) = \frac{\frac{1}{q}}{\sum_{h=2}^{m_{\max}} \frac{1}{h}} = \frac{\prod_{h=2}^{m_{\max}} h}{q \sum_{h=2}^{m_{\max}} \prod_{l=2, l \neq h}^{m_{\max}} h} \tag{21}$$

for all $q \in \{2, \ldots, m_{\max}\}$. This choice is made so that it is more likely to generate instances with limited numbers of partitions.

Thus, Algorithm 2 evaluates the solutions based on their objective functions and selects the best solution.

6. Simulation results

This section begins by demonstrating the correlation between the pairwise connectivity as defined by Equation (1) and the linear approximation $\widehat{PWC}$ introduced in Equation (16) in order to justify the adoption of the latter metric. Next, it illustrates the effectiveness of the integer linear programming optimization approach on an example network. Finally, the heuristic approach described in Algorithm 2 is applied to a real network with 332 nodes and 2,126 links representing U.S. airports and routes between the airports as of 1997.

6.1. Pairwise connectivity approximation

This section describes an experimental validation of the approximate pairwise connectivity index $\widehat{PWC}$ defined by Equation (16). This index is a linear combination of two of the three terms that constitute the objective function of the integer linear programming formulation: (i) number of connected components; and (ii) size of the largest connected component.

The validation strategy involved an analysis of the correlation between the pairwise connectivity and $\widehat{PWC}$ in an instance of a graph with $n = 40$ nodes. A total of 3,800 admissible solutions were sampled for a specific value of $\widehat{PWC}$, each solution corresponding to a pairwise connectivity that depended on the choice of α representing the trade-off between the two sub-objectives.

Figure 3: Correlations between PWC and $\widehat{PWC}$ for three α values.

Figure 3 shows the correlations between the pairwise connectivity and $\widehat{PWC}$ for three α values and 3,800 feasible instances. The correlations were obtained by selecting an α value and computing the pairwise connectivity and $\widehat{PWC}$ associated with each sampled solution. In order to consider realistic situations where a few nodes were attacked, the correlation values marked with gray asterisks are related only to the subset of gray points (i.e., corresponding to solutions with at most six attacked nodes). The correlation

values marked with black circles are related to all the sets of points (i.e., gray and black points, which have more than six attacked nodes).

The results in Figure 3 indicate that pairwise connectivity and $\widehat{PWC}$ appear to be tightly correlated – the correlation is always greater than or equal to $\rho = 0.82$. In particular, the correlation is larger in the case of a limited number of attacked nodes (e.g., the gray cloud of points in the figure). This is highly beneficial because the intent is to find inexpensive solutions in terms of attack cost that would be more representative of real attack strategies.

Figure 4: Correlations between PWC and $\widehat{PWC}$ for 21 α values.

Figure 4 further evaluates the ability of the $\widehat{PWC}$ index to closely approximate the pairwise connectivity. Specifically, the figure shows the correlations between the parameter α (which determines the trade-off between the two sub-objectives in $\widehat{PWC}$) and the correlation coefficient ρ between the pairwise connectivity and $\widehat{PWC}$. For each choice of α, 2,000 feasible solutions were generated for the same graph with $n = 40$ nodes. The correlations marked with black boxes are based on all the solutions while the correlations marked with gray triangles are associated with the subset of solutions with attacks targeting up to 15% of the nodes. The results reveal that the correlations, which are already high for small α values, tend to grow with α, reaching a plateau at 0.98 (gray line) and 0.90 (black line) for $\alpha \geq 0.5$.

6.2. Integer linear programming problem

This section demonstrates the application of the proposed integer linear programming formulation. Specifically, it analyzes the optimal solution to the formulation for a sample instance for different trade-offs involving the three objectives in the objective function.

(a) Optimal solution 1. (b) Optimal solution 2. (c) Optimal solution 3.

Figure 5: Optimal solutions of the integer linear programming formulation.

Figure 5 shows three optimal solutions for a graph with $n = 25$ nodes for 66 combinations of the parameters α_1, α_2 and α_3 considering an attack cost $k_i = 1$ for each node. The optimal solutions are associated with a small number of removed nodes (black) and a low rate of pairwise connectivity.

In Figure 5(a), the network is divided into three partitions after two nodes are attacked (and removed) ($PWC = 26\%$). In Figure 5(b), the network is divided into four partitions after three nodes are attacked ($PWC = 20\%$). In Figure 5(c), the network is divided into five partitions after four nodes are attacked ($PWC = 16.3\%$).

Figure 6: Critical nodes and pairwise connectivity for the optimal solutions in Figure 5.

Figure 6 shows the numbers of critical nodes (black circles) and the pairwise connectivity values (gray triangles) for the optimal solutions in Figure 5 for each triple of parameters α_1, α_2 and α_3; the specific combinations of the three parameters are shown in the stacked plot on the x-axis. As seen in Figure 6, as long as the value of α_1 is less than 0.4 (i.e., weight of the attack cost minimization objective matches the number of attacked nodes), the optimal solution involves a large number of critical nodes and, thus, yields a pairwise connectivity near zero. Therefore, these solutions barely describe the behavior of a real attacker.

Starting from the solutions associated with $\alpha_1 \geq 0.4$, the number of attacked nodes slowly decreases, while the pairwise connectivity increases. Among other combinations, Figure 5 shows the results of three choices of weights that correspond to the best solutions associated with the removal of two, three and four nodes, respectively (these attacked nodes are shown in black in Figure 5).

Table 1: Details of the optimal solutions.

$\boldsymbol{\alpha_1}$	$\boldsymbol{\alpha_2}$	$\boldsymbol{\alpha_3}$	$\boldsymbol{\lvert V_c \rvert}$	**PWC**
0.9	0.1	0	2	26%
0.6	0.4	0	3	20%
0.7	0	0.3	4	16.3%

Table 3 shows the details of the three optimal solutions in terms of the number of attacked nodes $|V_c|$ and the pairwise connectivity PWC. Note that the pairwise connectivity values in the three cases are quite similar (26%, 20% and 16.3%), although the parameters that produce the results in the three cases are considerably different. Conversely, the proposed formulation appears to be much more sensitive to the subtle trade-offs between the conflicting attacker objectives.

6.3. Case study: U.S. airport network

This section demonstrates the application of the heuristic approach described in Section 5 to obtain suitable solutions of the integer linear programming formulation in reasonable time. The results are compared against those obtained using another attack strategy from the research literature.

Figure 7: USAir97 network [4].

Figure 7 shows the U.S. airport network as it was in 1997; the corresponding USAir97 dataset is available at [4]. The network comprises 332 nodes and 2,126 edges. Each node represents an airport while each edge corresponds to a direct flight from one airport to another. The size of each node is proportional to its degree.

Figure 8: USAir97 network node-degree distribution.

Figure 8 shows the node-degree distribution of the USAir97 network (i.e., number of flight routes for each airport). The markers represent the degree frequency distribution and the solid line is the fitting curve. Most of the nodes are weakly connected to other nodes, but a small subset of nodes have a high node degrees and correspond to hubs.

In a real-world scenario, major airports are protected to a greater extent than minor airports. To model this fact, the attack cost is assumed to be proportional to the relevance of an airport. In other words, the attack cost of a node is equal to its degree. Thus, the attack cost for Chicago Airport (largest hub) is 139 while the cost associated with Abilene Regional Airport is one because it is only connected to Dallas airport.

The feasible solution generation algorithm was applied with $n_a = 8,000$ attempts; the maximum number of partitions was set to four and χ_{max} was set to 3,000. The objective function was evaluated for n_a solutions with ten different values of the weights α_1, α_2 and α_3 – this modeled ten different attack behaviors.

To analyze the effectiveness of the proposed attack strategy, the optimal solutions obtained were compared against a well-known attack strategy described in the literature. This strategy iteratively disconnects network nodes in the descending order of degree, which has been shown to be highly disruptive with regard to network connectivity (see, e.g., [13, 18]).

Table 2: Comparison of the proposed heuristic approach versus the degree-based attack strategy.

Budget	Heuristic Approach				Degree-Based Attack Strategy		
	Objective Function Weights	Attacked Nodes	Node IDs	PWC	Attacked Nodes	IDs	PWC
0.3123	α_1=0.8, α_2=0.1, α_3=0.1, α_1=0.9, α_2=0.1, α_3=0	1	117	**97.01**	1	330	**99.39**
0.3123	α_1=0.9, α_2=0, α_3=0.1	1	44	**98.79**	1	4	**99.39**
3.279	α_1=0.7, α_2=0.1, α_3=0.2	8	13, 75, 150, 153, 217, 237, 248, 268	**84.55**	1	290	**99.39**
4.7629	α_1=0.8, α_2=0.2, α_3=0, α_1=0.7, α_2=0.2, α_3=0.1, α_1=0.6, α_2=0.2, α_3=0.2, α_1=0.6, α_2=0.1, α_3=0.3	15	2, 13, 58, 75, 81, 117, 150, 153, 217, 237, 248, 268, 284, 305, 328	**80.71**	1	232	**97.60**
9.1355	α_1=0.6, α_2=0.3, α_3=0.1, α_1=0.6, α_2=0.4, α_3=0	16	2, 13, 58, 75, 81, 117, 150, 153, 217, 237, 248, 268, 273, 284, 305, 322	**74.97**	2	18 36	**97.60**

Table 2 and Figure 9 compare the results obtained with the proposed heuristic approach against those obtained with the degree-based strategy. Specifically, the heuristic approach was applied to ten $\alpha_1, \alpha_2, \alpha_3$ parameter combinations and the pairwise connectivity values were computed. Pairwise connectivity values were also computed for the degree-based strategy. The left-hand side of Table 3 presents the results of the heuristic approach in terms of cost (i.e., budget spent), objective function weights $(\alpha_1, \alpha_2, \alpha_3)$, numbers of attacked nodes and their IDs, and pairwise connectivity values. Although they use different weight triples, the ten attack strategies converge to the same targets and the same cost. Moreover, note that, when the value of α_1 decreases, the attack cost and the number of attacked nodes increase, and, consequently, the pairwise connectivity PWC decreases. The right-hand side of Table 3 shows the results of the degree-based attack strategy with the same budgets.

Figure 9: Comparison of pairwise connectivity values and budgets for the proposed heuristic approach (crosses) and the degree-based attack strategy (circles).

Each comparison assumed that the attacker has a fixed budget, which corresponds to the cost of the solution found by the heuristic approach. The degree-based attack strategy was then applied, which iteratively selected the nodes with the highest degrees until the budget was expended. Note that the budget and attack cost were normalized in the range from 0 to 332 to render them comparable with the attack cost in Equation (15).

An important observation is that, given a fixed budget, a large degradation in terms of pairwise connectivity is obtained by attacking several small nodes instead of a few hubs. Moreover, the selection of target nodes based on their relevance (i.e., node degree) produces a limited degradation in terms of pairwise connectivity. In contrast, when the proposed approach is applied, a significant degradation in pairwise connectivity is achieved by focusing the limited attacker resources on a set of small airports instead of large hubs. The results in the two PWC columns in Table 2 highlight the effectiveness of the proposed approach. Specifically, the strategy identified by the proposed approach divides the network into several partitions, which causes significant damage to network connectivity.

7. Conclusions

This paper has presented an integer linear programming optimization problem that finds the critical nodes in a network – specifically the nodes whose removal have severe impacts on network connectivity. The novelty of the approach lies in the adoption of an attacker perspective that incorporates the conflicting objectives of minimizing the ability of nodes to communicate with each other and minimizing the attack cost. The objectives are mediated by weights that model various attacker preferences. Unlike other related research, no assumptions are made about the number of nodes that are attacked and the number of partitions existing after an attack. Moreover, the proposed formulation is much more descriptive while maintaining lower complexity, rendering it very useful for predicting attacks and identifying the nodes that must be protected. Since exact solutions of the formulation are computationally expensive for large problems, a heuristic algorithm is presented to obtain an approximate solution. Simulation results using the USAir97 airport network dataset demonstrate the effectiveness and utility of the heuristic approach.

Modeling attacker behavior and identifying critical assets constitute the first step in network infrastructure protection. Future research will focus

on scenarios where an attacker encounters different, possibly dynamically-changing, costs for attacking different nodes; and on enabling decision makers to identify the nodes that must be protected. The proposed framework will also be augmented to introduce an optimization problem for implementing defenses. In this case, it will be necessary to cast the resulting coupled optimization problems using game-theoretic concepts in order to obtain optimal solutions that achieve equilibrium.

Acknowledgement

This research was partially supported by the SECUREWATER Project, which was funded by the Italian Ministry of Foreign Affairs and International Cooperation.

IMPORTANT NOTE TO IJCIP TYPESETTERS: I have checked and edited the references in this paper myself. Please DO NOT MODIFY the references – except to add hyperlinks. Please contact the Journal Manager Ms. Begum Salma Pattan if you have any questions.
Professor Sujeet Shenoi, Editor-in-Chief, IJCIP

References

[1] R. Albert, H. Jeong and A. Barabasi, Error and attack tolerance of complex networks, *Nature*, vol. 406, pp. 378–382, 2000.

[2] A. Arulselvan, C. Commander, L. Elefteriadou and P. Pardalos, Detecting critical nodes in sparse graphs, *Computers and Operations Research*, vol. 36(7), pp. 2193–2200, 2009.

[3] A. Arulselvan, C. Commander, O. Shylo and P. Pardalos, Cardinality-constrained critical node detection problem, in *Performance Models and Risk Management in Communications Systems*, N. Gulpinar, P. Harrison and B. Rustem (Eds.), Springer, New York, pp. 79–91, 2011.

[4] V. Batagelj and A. Mrvar, Pajek Datasets (`vlado.fmf.uni-lj.si/pub/networks/data`), 2006.

[5] Y. Berezin, A. Bashan, M. Danziger, D. Li and S. Havlin, Localized attacks on spatially embedded networks with dependencies, *Scientific Reports*, vol. 5, article no. 8934, 2015.

[6] G. Brown, W. Carlyle, J. Salmeron and K. Wood, Analyzing the vulnerability of critical infrastructure to attack and planning defenses, in *Tutorials in Operations Research: Emerging Theory, Methods and Applications*, H. Greenberg and J. Smith (Eds.), Institute for Operations Research and Management Science, Hanover, Maryland, pp. 102–123, 2005.

[7] G. Brown, W. Carlyle, J. Salmeron and K. Wood, Defending critical infrastructure, *Interfaces*, vol. 36(6), pp. 530–544, 2006.

[8] P. Crucitti, V. Latora and M. Marchiori, Locating critical lines in high-voltage electrical power grids, *Fluctuation and Noise Letters*, vol. 5(2), pp. L201–L208, 2005.

[9] T. Dinh, Y. Xuan, M. Thai, P. Pardalos and T. Znati, On new approaches for assessing network vulnerability: Hardness and approximation, *IEEE/ACM Transactions on Networking*, vol. 20(2), pp. 609–619, 2012.

[10] M. Di Summa, A. Grosso and M. Locatelli, Branch and cut algorithms for detecting critical nodes in undirected graphs, *Computational Optimization and Applications*, vol. 53(3), pp. 649–680, 2012.

[11] European Commission, The Directive on Security of Network and Information Systems (NIS Directive), Strasbourg, France, 2016.

[12] L. Faramondi, G. Oliva, F. Pascucci, S. Panzieri and R. Setola, Critical node detection based on attacker preferences, *Proceedings of the Twenty-Fourth Mediterranean Conference on Control and Automation*, pp. 773–778, 2016.

[13] P. Holme, B. Kim, C. Yoon and S. Han, Attack vulnerability of complex networks, *Physical Review E: Statistical, Nonlinear and Soft Matter Physics*, vol. 65(5), article no. 056109, 2002.

[14] X. Huang, J. Gao, S. Buldyrev, S. Havlin and H. Stanley, Robustness of interdependent networks under targeted attacks, *Physical Review E:*

Statistical, Nonlinear and Soft Matter Physics, vol. 83(6), article no. 065101, 2011.

[15] M. Lalou, M. Tahraoui and H. Kheddouci, Component-cardinality-constrained critical node problem in graphs, *Discrete Applied Mathematics*, vol. 210, pp. 150–163, 2016.

[16] O. Lordan, J. Sallan, P. Simo and D. Gonzalez-Prieto, Robustness of the air transport network, *Transportation Research Part E: Logistics and Transportation Review*, vol. 68, pp. 155–163, 2014.

[17] V. Louzada, F. Daolio, H. Herrmann and M. Tomassini, Generating robust and efficient networks under targeted attacks, in *Propagation Phenomena in Real World Networks*, D. Król, D. Fay and B. Gabrys (Eds.), Springer International Publishing, Cham, Switzerland, pp. 215–225, 2015.

[18] Z. Lu and X. Li, Attack vulnerability of network controllability, *PLOS ONE*, vol. 11(9), e0162289, 2016.

[19] W. Pullan, Heuristic identification of critical nodes in sparse real-world graphs, *Journal of Heuristics*, vol. 21(5), pp. 577–598, 2015.

[20] S. Shao, X. Huang, H. Stanley and S. Havlin, Percolation of localized attacks on complex networks, *New Journal of Physics*, vol. 17(2), article no. 023049, 2015.

[21] Y. Shen, N. Nguyen, Y. Xuan and M. Thai, On the discovery of critical links and nodes for assessing network vulnerability, *IEEE/ACM Transactions on Networking*, vol. 21(3), pp. 963–973, 2013.

[22] G. Stergiopoulos, P. Kotzanikolaou, M. Theocharidou and D. Gritzalis, Risk mitigation strategies for critical infrastructures based on graph centrality analysis, *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 34–44, 2015.

[23] F. Sun and M. Shayman, On pairwise connectivity of wireless multihop networks, *International Journal of Security and Networks*, vol. 2(1-2), pp. 37–49, 2007.

[24] M. Ventresca, K. Harrison and B. Ombuki-Berman, An experimental evaluation of multi-objective evolutionary algorithms for detecting critical nodes in complex networks, *Proceedings of the Eighteenth European Conference on the Applications of Evolutionary Computation*, pp. 164–176, 2015.

[25] S. Wang, L. Hong, M. Ouyang, J. Zhang and X. Chen, Vulnerability analysis of interdependent infrastructure systems under edge attack strategies, *Safety Science*, vol. 51(1), pp. 328–337, 2013.

[26] J. Wu, H. Deng, Y. Tan and D. Zhu, Vulnerability of complex networks under intentional attacks with incomplete information, *Journal of Physics A: Mathematical and Theoretical*, vol. 40(11), pp. 2665–2671, 2007.

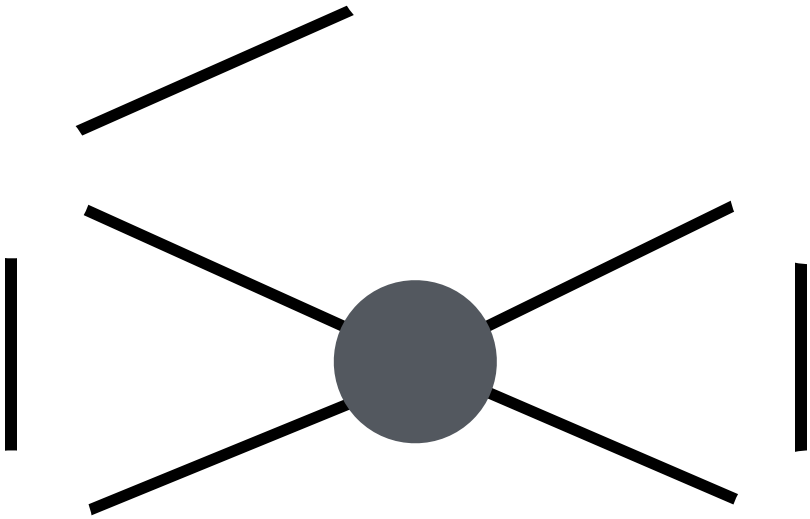
[27] Z. Zhang, X. Li and H. Li, A quantitative approach for assessing the critical nodal and linear elements of a railway infrastructure, *International Journal of Critical Infrastructure Protection*, vol. 8, pp. 3–15, 2015.

Algorithm 1 : Feasible solution generation (FSG).

$c_j = 0, \forall j = 1, \dots n$
$x_j^{(i)} = 0, \forall i = 1, \dots, m$ and $\forall j = 1, \dots n$
$t_i = 0, \forall i = 1, \dots, m$
$q = 0$
$\phi = 0$ ▷ non-empty partitions
$\chi = 0$ ▷ no node has been reconsidered
for $i = 1 \dots n$ **do**
    $\mathcal{M}_i = \emptyset$ ▷ assigned neighbors of v_i
    $p(i) = 0$ ▷ v_i is not assigned
end for
$\mathcal{I} = V$
while $\mathcal{I} \neq \emptyset$ and $\chi < \chi_{max}$ **do**
    select random $v_i \in \mathcal{I}$
    $\mathcal{I} = \mathcal{I} \backslash \{v_i\}$
    ▷ if no neighbor of v_i has been assigned
    if $\mathcal{M}_i = \emptyset$ **then**
        ▷ if an empty partition exists, attempt to assign v_i to a random empty partition; otherwise, reconsider the node v_i later to avoid too many critical nodes
        if $\phi < m$ **then**
            select random $h \in \{1, \dots, m\}$ such that $V_h = \emptyset$
            $p(i) = h$
            $\phi = \phi + 1$
            $\mathcal{M}_j = \mathcal{M}_j \cup \{v_i\}, \forall v_j \in \mathcal{N}_i$
        else
            $\mathcal{I} = \mathcal{I} \cup \{v_i\}$ ▷ replace v_i in $\mathcal{I}$
            $\chi = \chi + 1$ ▷ increase the reconsiderations
        end if
    else
        ▷ if all the already-assigned neighbors of v_i are in the same partition h, assign v_i to partition h; otherwise, assign v_i to the set of critical nodes
        if $p(j) = h$ for some $h > 0$ and $\forall v_j \in \mathcal{M}_i$ **then**
            $p(i) = h$
            $\mathcal{M}_j = \mathcal{M}_j \cup \{v_i\}, \forall v_j \in \mathcal{N}_i$
        else
            ▷ do nothing; critical nodes are assigned at the end
        end if
    end if
end while
▷ choose assignment variables
for $i = 1 \dots n$ **do**
    $x_i^{(p_i)} = 1$
end for
▷ assign critical nodes
for $j = 1 \dots n$ **do**
    $c_j = 1 - \sum_{i=1}^{m} x_j^{(i)}$
end for
▷ select t_i
for $i = 1 \dots m$ **do**
    if $|x^{(i)}| > 0$ **then**
        $t_i = 1$
    end if
end for
$q = \max\{|x^{(1)}|, \dots, |x^{(m)}|\}$
return $y = [c_1, \dots, c_n, x_1^{(1)}, \dots, x_n^{(n-1)}, q, t_1, \dots, t_{n-1}]^T$

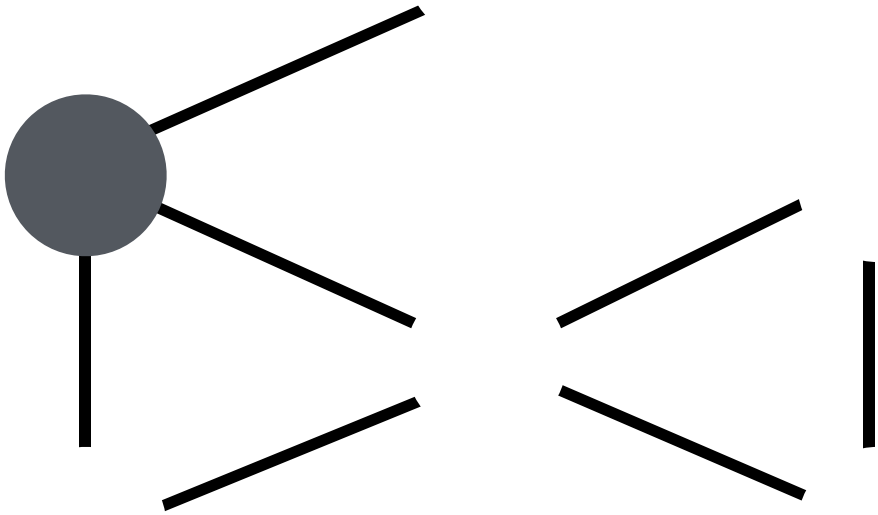
Algorithm 2 : Heuristic network vulnerability detection.

$y_{\max} = \emptyset$
$z_{\max} = \infty$
for $i = 1, \ldots, n_a$ **do**
  Select $m_{tmp} \in \{2, \ldots, m_{\max}\}$ with probability specified by Equation (21)
  $y_{tmp} = FSG(G, m_{tmp})$
  if $r^T y_{tmp} < z_{\max}$ **then**
    $z_{\max} = r^T y_{tmp}$
    $y_{\max} = y_{tmp}$
  end if
end for
return $y_{\max}$



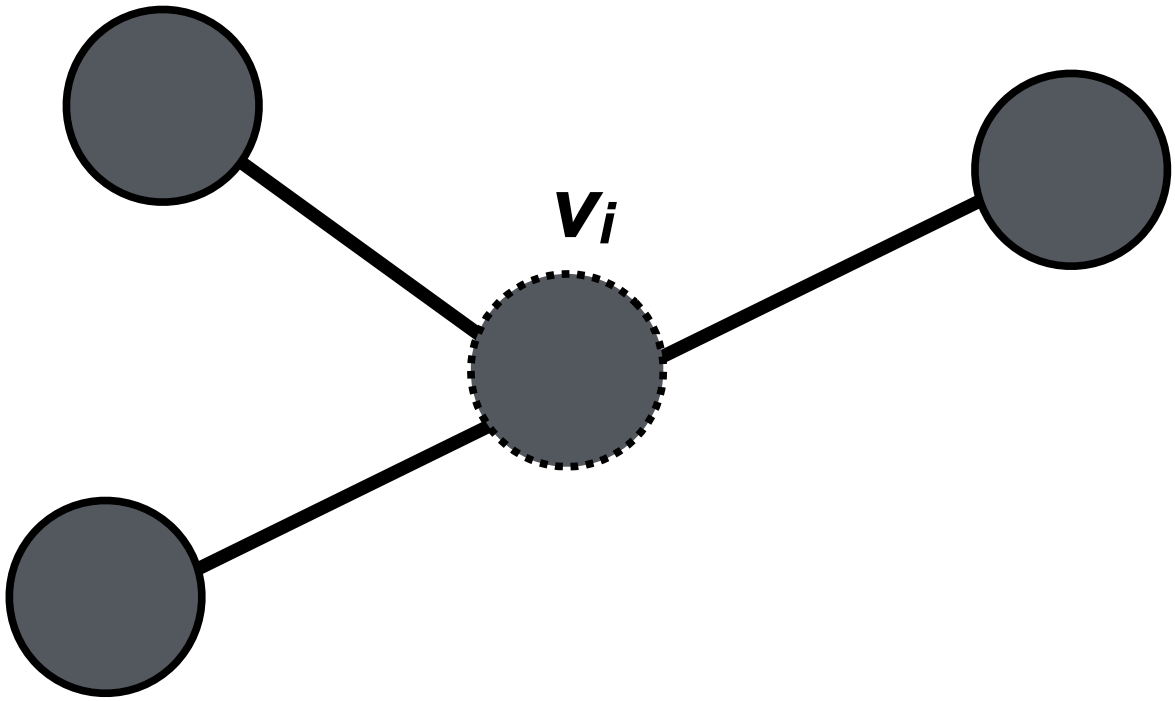
(a) $PWC = 26.6\%$.

Figure 1. Node deletion in central and peripheral areas.



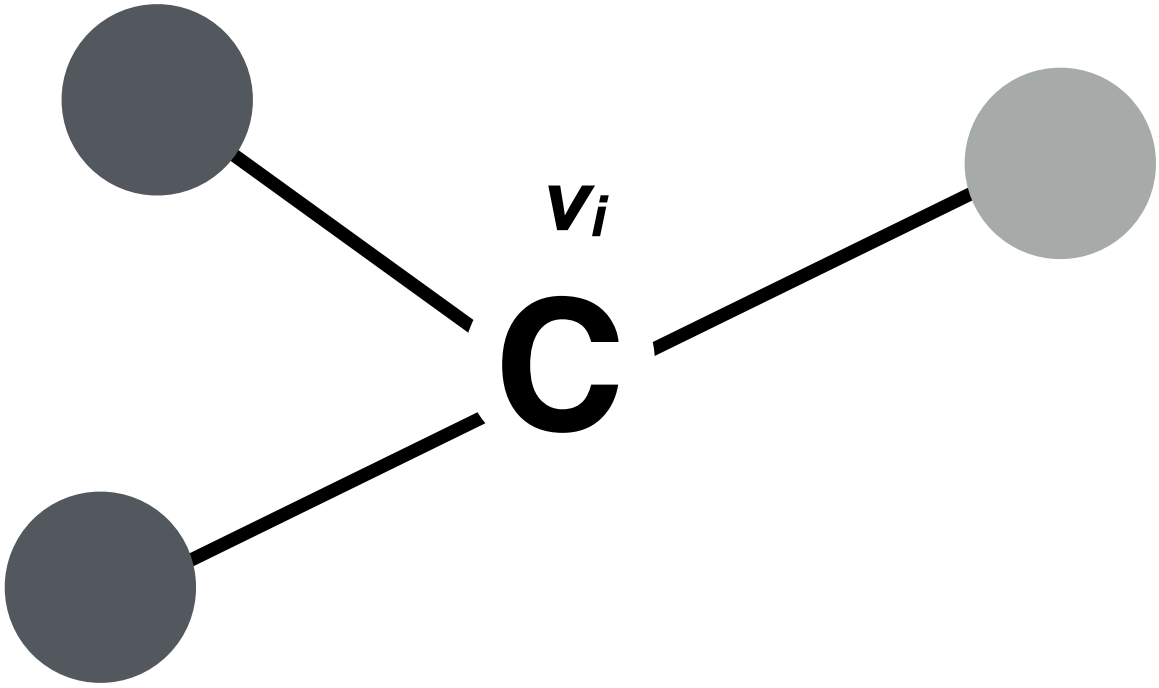
(b) $PWC = 40\%$.

Figure 1. Node deletion in central and peripheral areas.



(a) Feasible solution 1.

Figure 2. Heuristic assignment criterion adopted by Algorithm 1.



(b) Feasible solution 2.

Figure 2. Heuristic assignment criterion adopted by Algorithm 1.

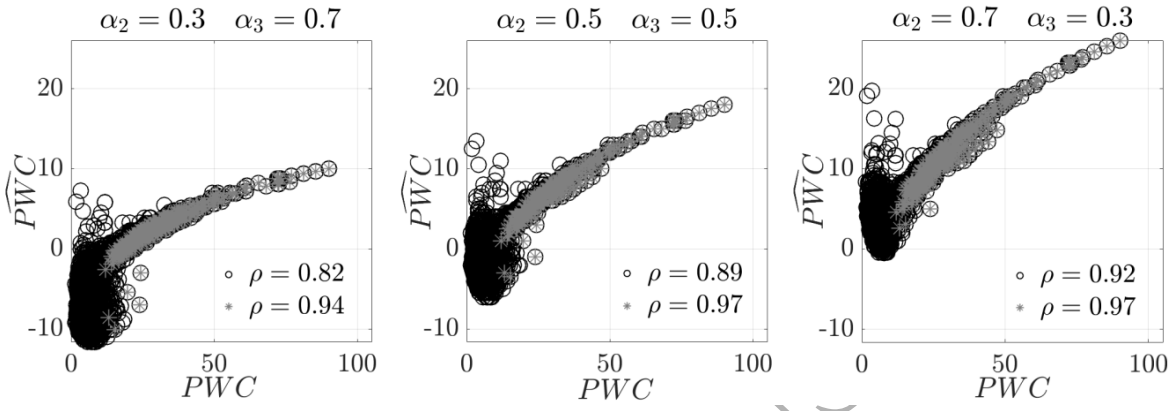


Figure 3. Correlations between PWC and $\widehat{PWC}$ for three α values.

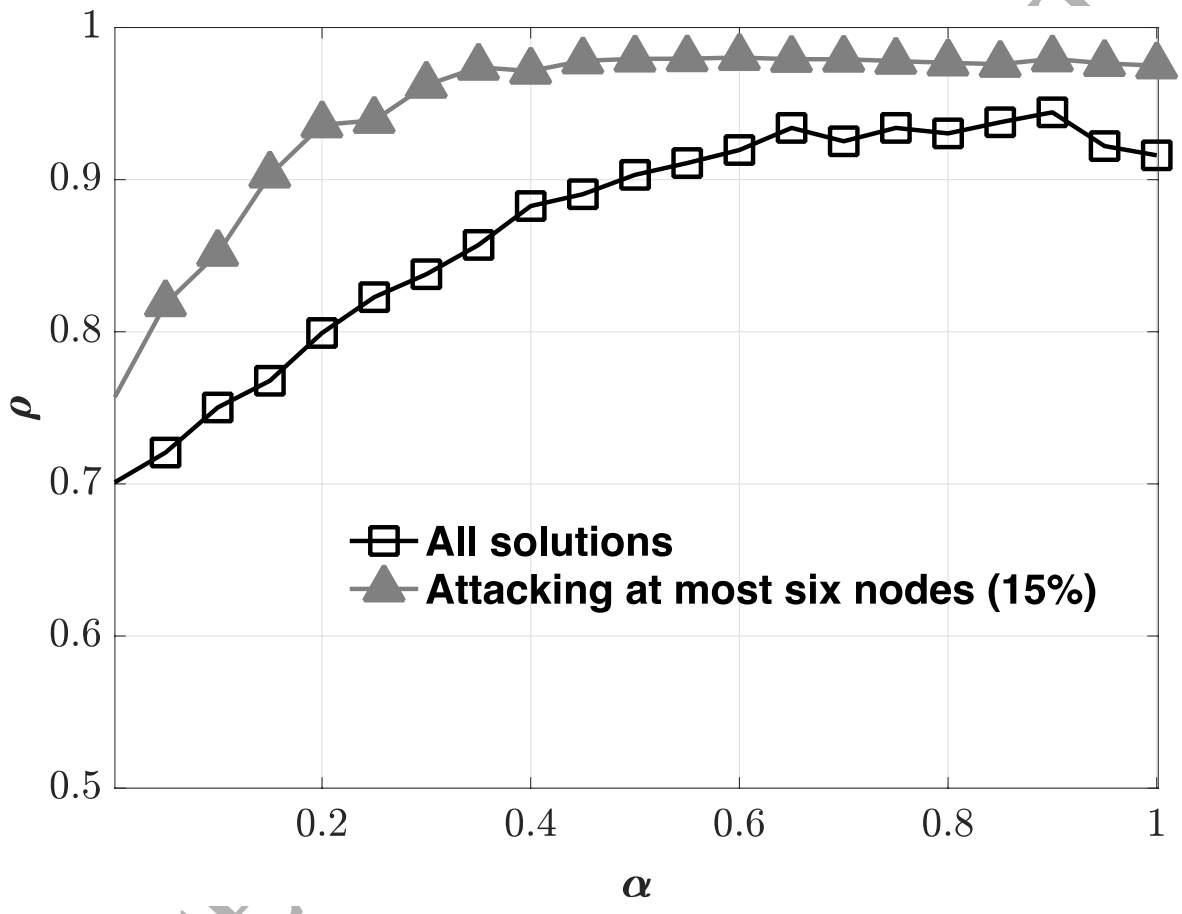
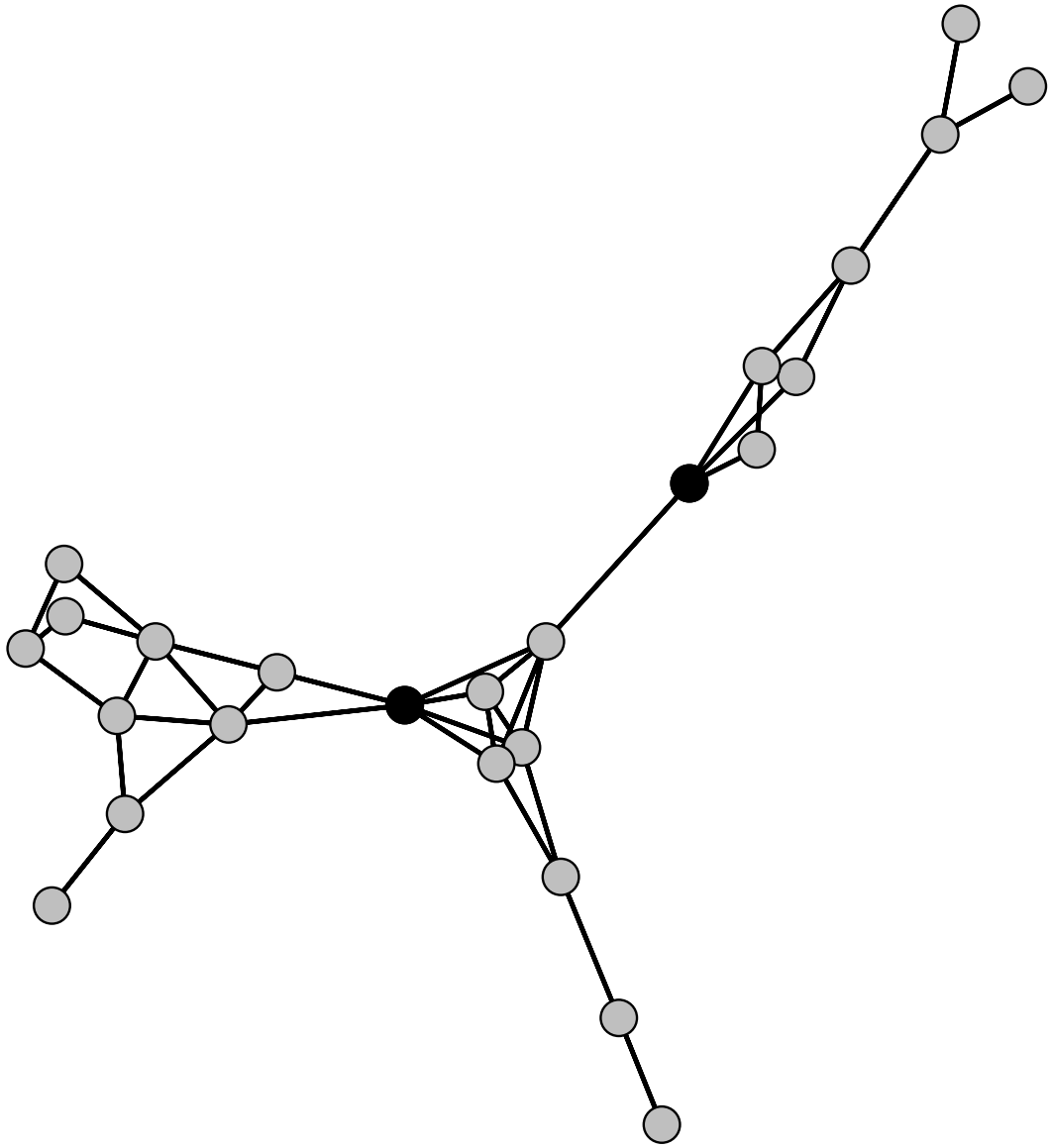
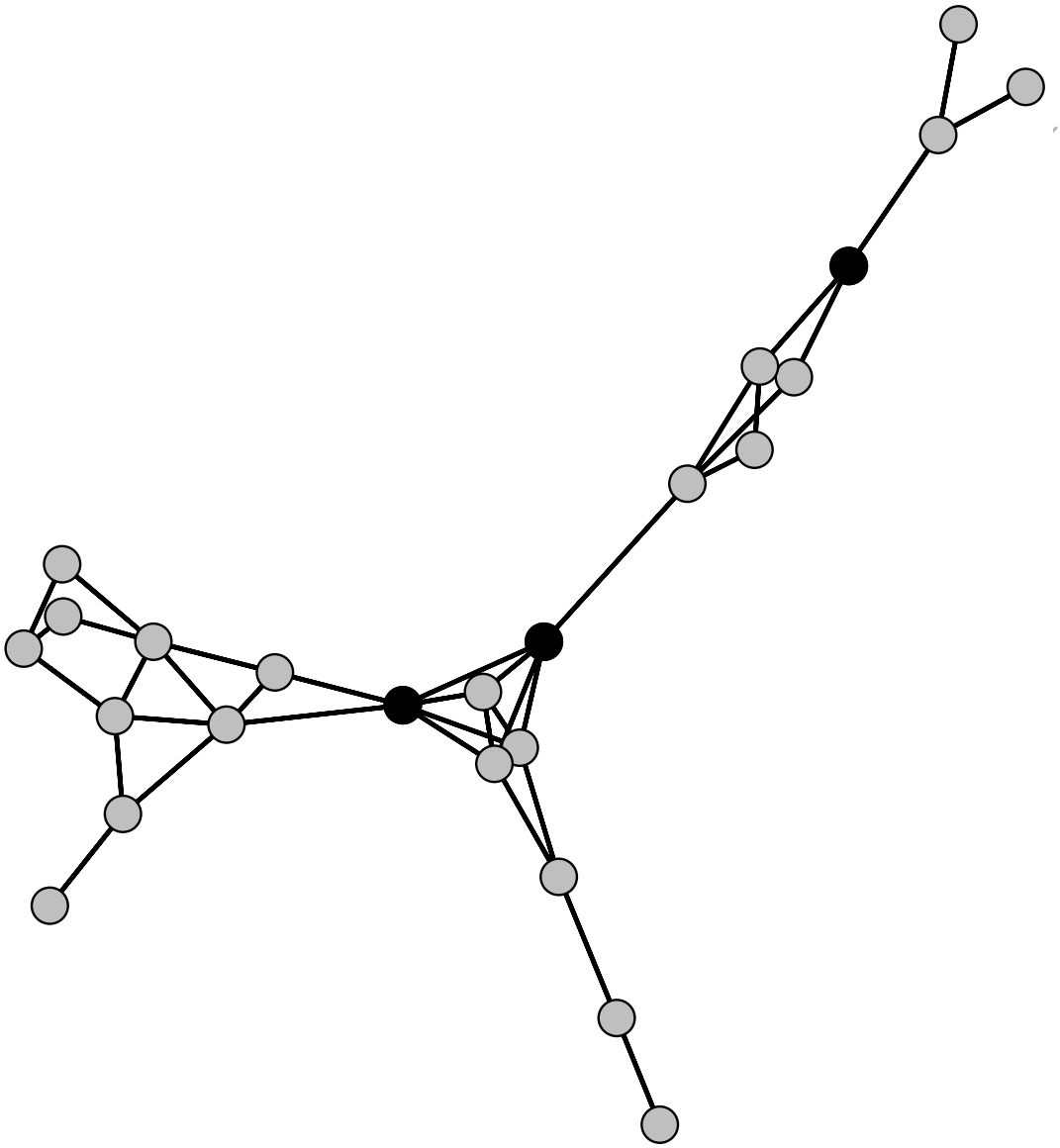


Figure 4. Correlations between PWC and $\widehat{PWC}$ for 21 α values.



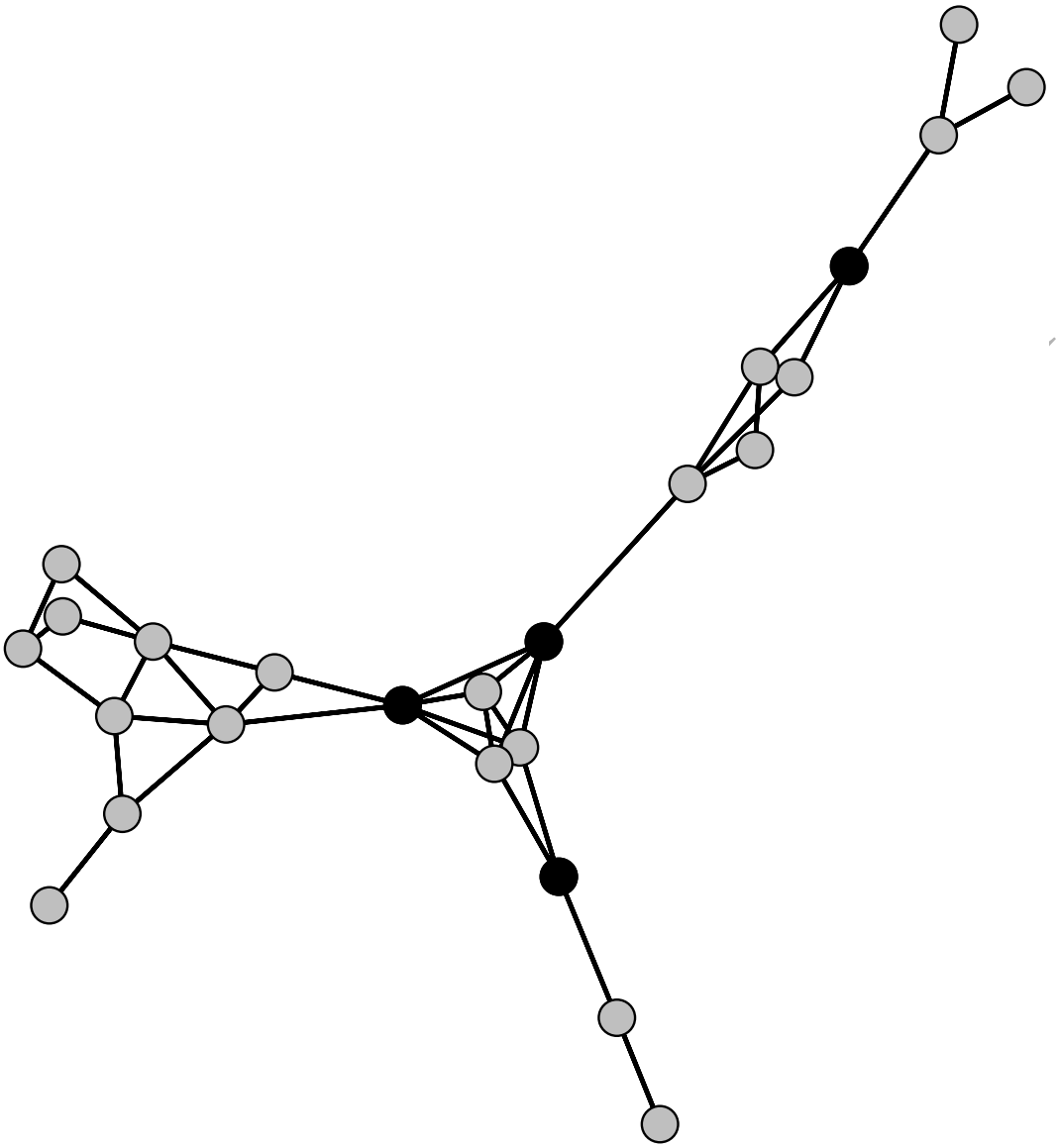
(a) Optimal solution 1.

Figure 5. Optimal solutions of the integer linear programming formulation.



(b) Optimal solution 2.

Figure 5. Optimal solutions of the integer linear programming formulation.



(c) Optimal solution 3.

Figure 5. Optimal solutions of the integer linear programming formulation.

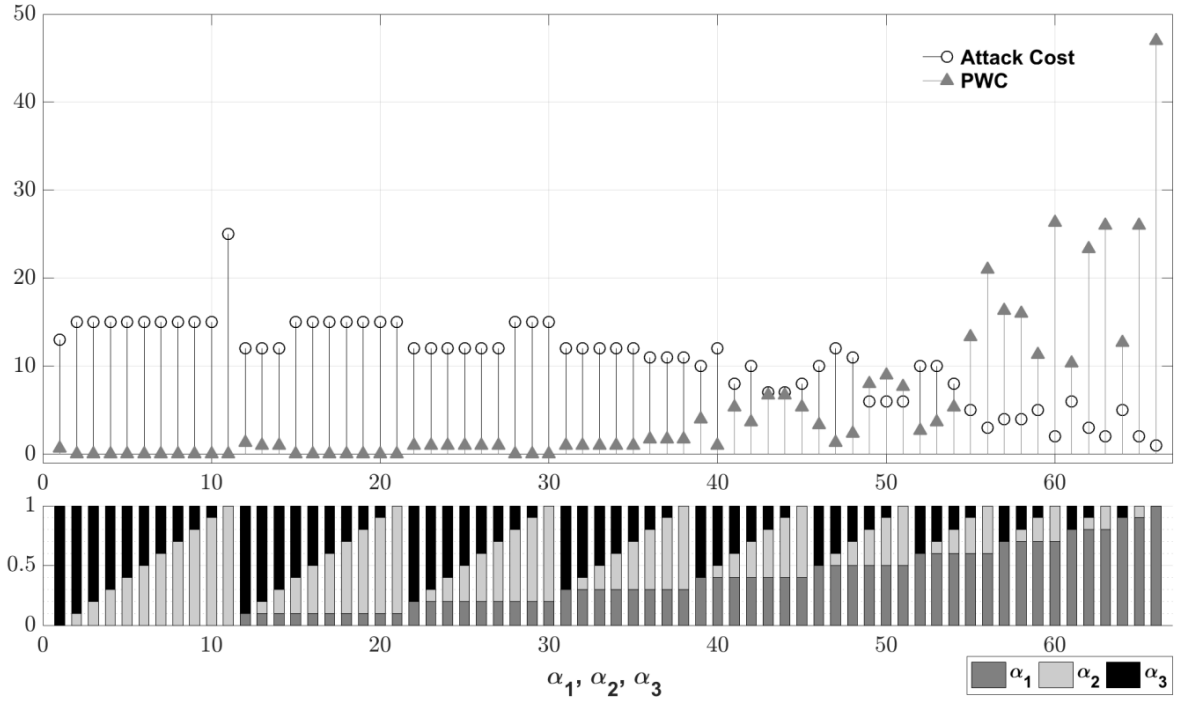


Figure 6. Critical nodes and pairwise connectivity for the optimal solutions in Figure~5.

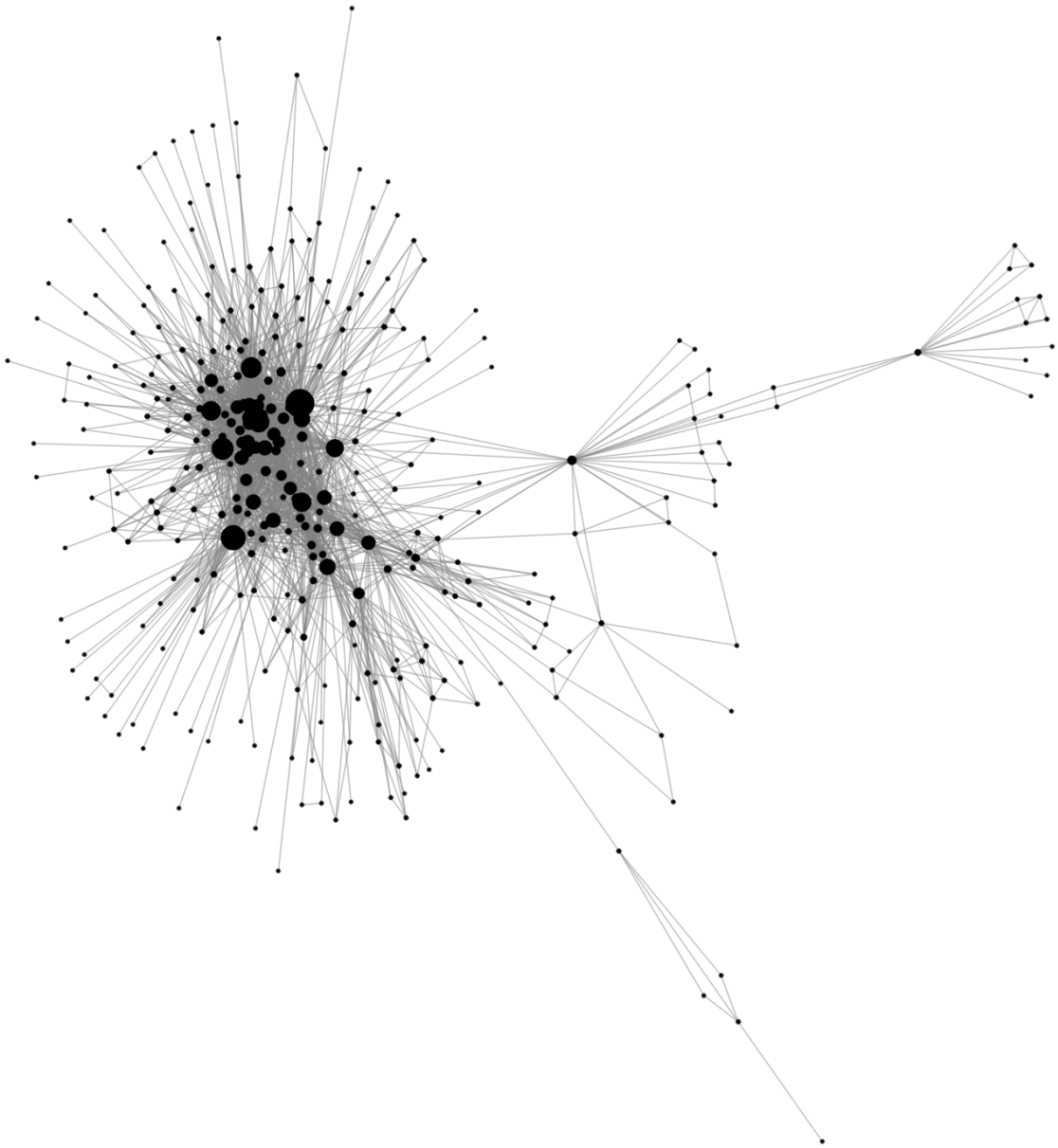


Figure 7. USAir97 network~\cite{usair97}.

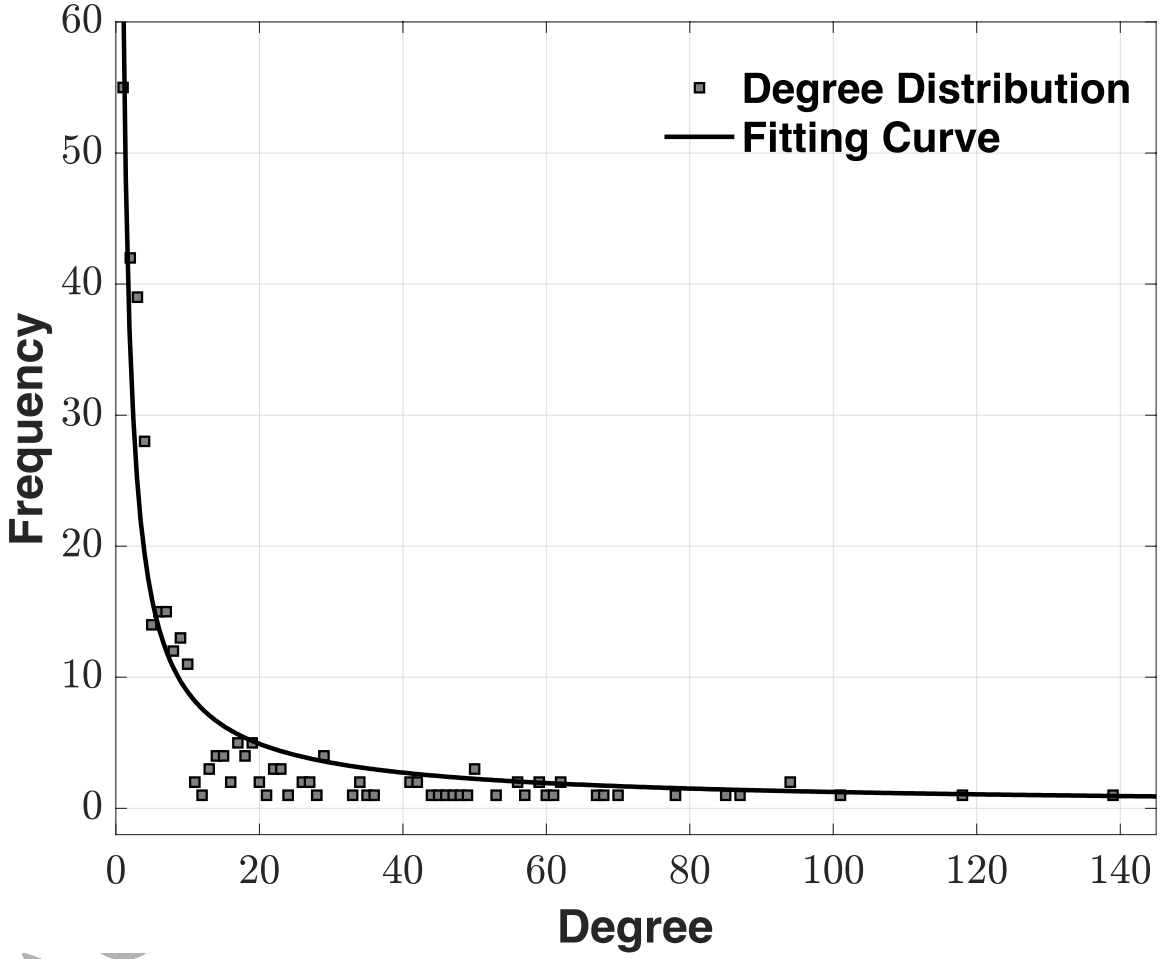


Figure 8. USAir97 network node-degree distribution.

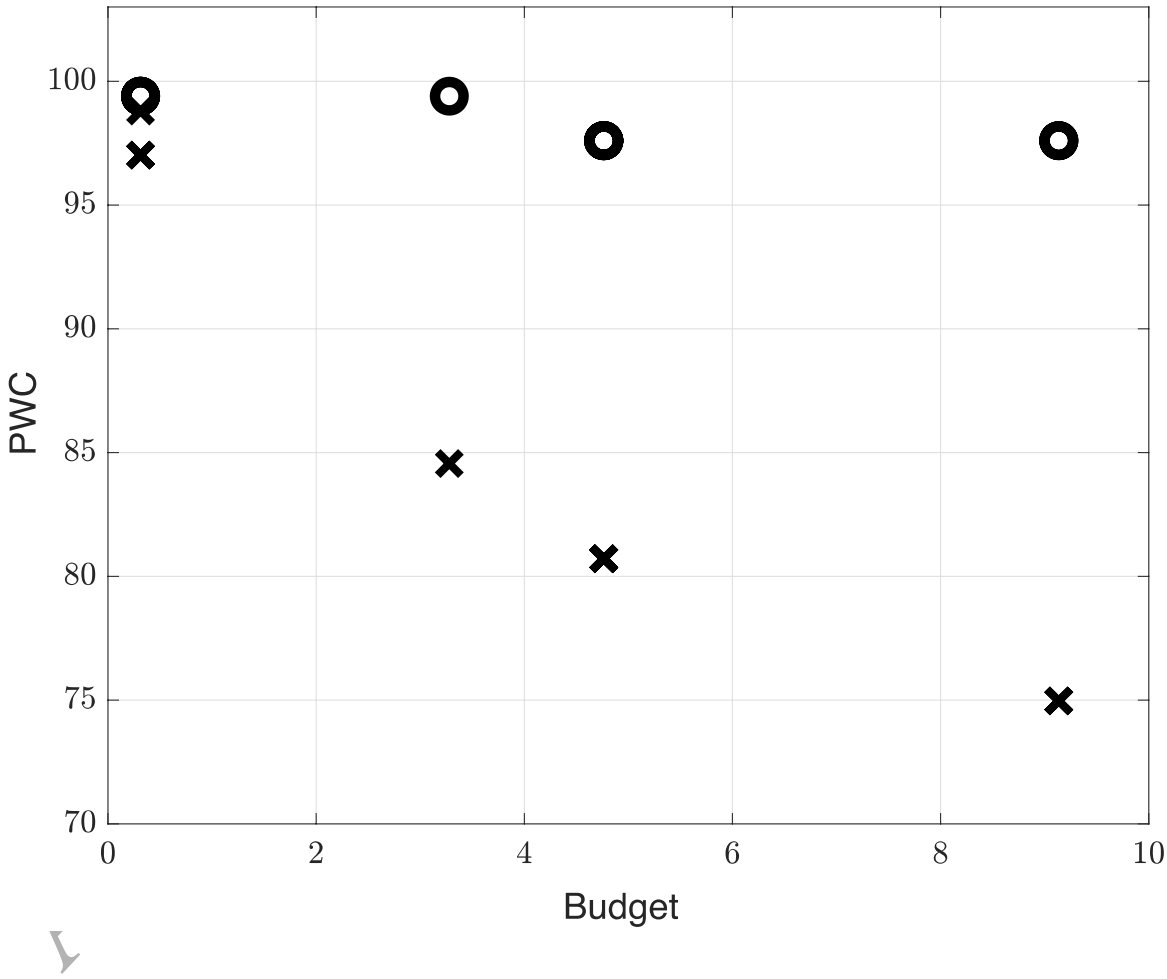


Figure 9. Comparison of pairwise connectivity values and budgets for the proposed heuristic approach (crosses) and the degree-based attack strategy (circles).