Finding Critical Elements in Infrastructure Networks

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

[a] *University Rome Tre, Department of Engineering, Via della Vasca Navale 79, 00146, Rome, Italy.*
[b] *Unit of Automatic Control, Department of Engineering, Università Campus Bio-Medico di Roma, via Álvaro del Portillo 21, 00128, Rome, Italy.*
[c] *Corresponding author. Email l.faramondi@unicampus.it*

Abstract

It is well known that profiling the attacker behavior is an effective way to obtain insights on network vulnerabilities and to identify locations that need to be protected. In this paper we present a novel Integer Linear Programming (ILP) formulation to model strategy of an attacker who targets a set of nodes (e.g., destroying or compromising them). To this end, we model attackers aiming to both deal the largest possible damage and minimize the attacker's effort. Specifically, we assume that the attacker is guided by three conflicting objectives: minimize the attack cost, maximize the number of disconnected components and minimize the size of the largest connected component. With respect to the state of the art, the proposed formulation is remarkably more descriptive, while keeping lower complexity; thus, it represents a valuable tool to predict attacks and to identify locations that need to be protected. However, since the exact solution of the formulation turns can be computationally expensive for large instances, we complement the paper by providing an heuristic algorithm to find an approximated solution. A simulation campaign, whose goal is to show the potential of the proposed approach, concludes the paper.

Keywords: Network Vulnerabilities, Critical Nodes, Attacker Perspective, Attacker Profiling

2010 MSC: 00-01, 99-00

1. Introduction

Since the seminal works of Albert et al. [1] and Holme et al. [2] early in 2000s, it has become evident that those attacks taking into account the topological structure of a network (e.g., telecommunication networks, road infrastructures, or other systems) may have dramatic consequences. Indeed, by knowing the topology of the network, an attacker can select the target sites more effectively, increasing the damages (e.g., in terms of disconnection of large portions of the network) while keeping the cost of the attack at a minimum (see [3, 4, 5, 6, 7] for recent works on the topic).

In this paper, we characterize the behavior of an attacker that targets some of the nodes in a network, by corrupting or disrupting them. We assume that the attacker objective is to cause the maximum damage with the minimum attack cost.

We formalize the problem in terms of *Integer Linear Programming* (ILP), by introducing suitable constraints that will be discussed later. Moreover, we provide an heuristic algorithm to find a sub-optimal solution in a cost-effective way. The main contribution of this paper is to provide a methodology to identify critical sites to be protected by considering the preferences of the attacker with respect to the different clashing objectives (i.e., maximize damage dealt and minimize the cost of the attack).

We point out that such an attacker profiling represents an effective way to identify and manage Critical Infrastructures' vulnerabilities and a first step in the implementation of possible risk mitigation strategies. This task is extremely important, as highlighted for instance by the NIS EU Directive [8], which requires to Critical Infrastructure operators, and specifically to IT providers, to take adequate measures in order to manage risk, report security incidents to the national competent authorities and provide early threat warnings.

1.1. Related Work

In recent years, techniques based on operational research and graph theory have been extensively applied in the context of the critical infrastructure protection. Among other approaches, it is worth mentioning methods to estimate and analyze the resilience in infrastructure networks that are based on (constrained) optimization problems [12, 13, 14, 15, 16, 18, 19, 20], bi-level optimization frameworks [23, 24] and network spectral analysis [22, 25, 26, 27, 28]. In particular, we point out that several of such approaches model the infrastructures in terms of a graph and evaluate its robustness by identifying *critical nodes*.

Other solutions in the literature encompass solutions based on the concept of *critical links* [10, 11, 15], i.e., links that once removed may cause a relevant degradation of some connection-related index such as the average inverse geodesic length (i.e., the sum of the inverse of the shortest paths among any pair of nodes) or the total pairwise connectivity [9].

Due to the intrinsic high computational complexity associated to such approaches, critical nodes are often detected via graph spectral analysis (see, for instance [25] where percolation theory is used to assess network robustness). In this case, metrics such as the degree of the nodes or the eigenvalues of the adjacency matrix are studied in order to evaluate the network robustness. In [22], the problem to estimate the network vulnerability is addressed by evaluating the structural controllability of the network after iteratively removing the nodes in order of degree, eigenvector centrality, and betweenness. We point out that such metrics have been often adopted in the context of Critical Infrastructures Protection (see [27, 28] and references therein), in order to assess node criticality and to provide risk mitigation strategies.

We point out that these latter approaches are generally developed and adopted in the context of electric networks. These approaches take into account the capabilities of a network to remain connected without considering specific source-destination links or specific paths. This aspect is handled also in the framework of *Critical Node Disruptor* problem (CND) [12, 13, 15]. Within CND, an attacker removes some of the nodes in the network with the aim to minimize the *total pairwise connectivity* [9], that is, the amount of pairs of nodes that are connected by a path after the removal of the attacked nodes. However, such a formalism, assumes that the attacker has either a priori knowledge on the maximum number k of nodes that have to be disconnected or the ability to disconnect up to a fixed number k of nodes; an incorrect choice of k may result in infeasible or inefficient solutions. Such approaches appears suitable to estimate the effectiveness of an attack on communication and transportation network where it is relevant to consider the capability of a source to reach a specific destination. In [17], a dual problem is addressed, namely *Cardinality Constrained Critical Node Detection Problem*, where a maximum allowed connected graph component size is specified and the objective is to minimize the number of attacked nodes required to fulfill this constraint. In [18], it is argued that the CND problem is intrinsically a multi-objective problem, and an improvement is suggested where not only the pairwise connectivity is minimized, but also the variance in cardinality among the connected components, i.e., the dimension of the "islands" obtained after the removal of the k nodes; anyhow such an approach suffers the same drawbacks of the standard CND problem. The approach in [19] is structured on a similar line, indeed, the size of each obtained connected component is constrained to be below a given bound. In [20], we provide a formulation in which the attacker is not constrained to attack a fixed number of nodes. Indeed, he/she aims at dividing the network in a fixed number of components while having two conflicting objectives: minimizing the number of attacked nodes and the size of the largest component.

1.2. Contribution

In this paper we propose an optimization formulation to solve the CND problem that overcomes some of the limitations of previous approaches. In our formulation, the attacker tries to obtain a large impact with limited resources. To this end he/she aims at finding nodes, which, if removed, disconnect the network into a large number of small partitions; in this way, each node is able to communicate only with a small subset of nodes. In particular, we do not require the user to specify a fixed number of partitions; the maximization of the number of disconnected components becomes an additional objective that has to be mediated with the minimization of the attack cost and the minimization of the size of the largest component. Indeed, the proposed formulation represents a relevant improvement with respect to previous CND approaches. In fact, by removing the constraint concerning fixed parameters such as the number of partitions [20] the attack cost [12] or the size of the largest partition [14], we are able to better reproduce the behavior of an attacker by shifting the focus to his/her preferences, without making assumption about the features of the final solution. However we point out that, similarly to previous literature [15], the proposed approach requires $O(n^2)$ Boolean decision variables, where n is the number of nodes in the network. Therefore, finding the exact solution of the proposed formulation might still be quite computationally expensive. To tackle this issue, we develop an heuristic algorithm able to provide good approximated solutions in reasonable time. The effectiveness of the proposed approach is verified with respect to a real-world network.

1.3. Paper Outline

The outline of the paper is as follows: we introduce some preliminary notations and definitions in Section 2; in Section 3 we develop the proposed ILP formulation while in Section 4 we describe the proposed heuristic algorithm; we report the results of a simulation campaign aimed at highlighting the potential of the proposed approach in Section 5; finally, we discuss some conclusive remarks and we discuss future work directions in Section 6.

2. Preliminaries

2.1. General Preliminaries

In the following we denote by $|X|$ the cardinality of a set X; vectors are represented via boldface letters, and we use $\mathbf{k}_m$ to indicate a vector in $\mathbb{R}^m$ whose components are all equal to k. We denote by $0_{n,m}$ an $n \times m$ matrix whose entries are all 0, and by I_n the $n \times n$ identity matrix.

Let an $n \times m$ matrix A and a $p \times q$ matrix B; the *Kronecker product* of A and B is the $np \times mq$ matrix

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

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

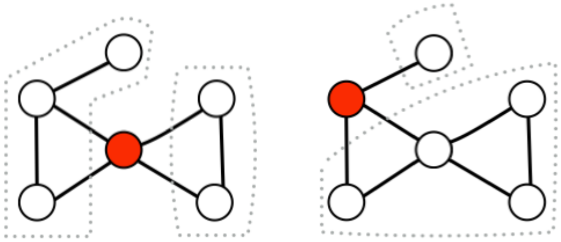
2.2. Graph-related Preliminaries

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 $(v_i, v_j) \in E$ captures the existence of a relation between node v_i and node v_j. A *partition* $V_i \subseteq V$ is a subset of the nodes in V.

A graph is said to be *undirected* if $(v_i, v_j) \in E$ whenever $(v_j, v_i) \in E$, and it is said to be *directed* otherwise. In the remainder of this paper we will only consider undirected graphs.

A *path* over a graph $G = \{V, E\}$, starting at a node $v_i \in V$ and ending at a node $v_j \in V$, is a subset of links in E that connects v_i and v_j, respecting the edge orientation and without creating loops. The length of a path is the number of links that compose it, while a *minimum path* is a path of minimum cardinality. An undirected graph is *connected* if for each pair of nodes v_i, v_j there is a path over G that connects them. 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 $x = (v_i, v_j) \in E$



(a) $PWC_{\%} = 26.6\%$ (b) $PWC_{\%} = 40\%$

Figure 1: Node deletion in central and peripherals area.

and a node y it holds

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

Note that 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) [9] 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 $p(v_i, v_j)$ is 1 if the pair (v_i, v_j) is connected via a path over G, and 0 otherwise. The PWC is a measure of connectivity that takes into account the existence of a path among any pair of nodes; the maximum $PWC(G) = 1$ is attained whenever the graph is connected. In the following we express the PWC in percentage, i.e., we consider $PWC_{\%}(G) = 100PWC(G)$.

Note that, for undirected graphs, the PWC 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 G, there are exactly $|V_i|(|V_i| - 1)/2$ distinct pairs of connected nodes. Assuming that G contains m connected components, we can alternatively express Equation (1) as

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

In Figure 1 we show that the deletion of different nodes may have remarkably different effects in terms of PWC. In Figure 1a, by removing the central node, the graph is disconnected in two components and only four pairs of nodes are connected by a path; as a result, $PWC_{\%} = 26.6\%$. In Figure 1b the graph is decomposed in two components again, but in this case six pairs of nodes are connected by a path and $PWC_{\%} = 40\%$. The above figure suggests that disconnected graphs whose connected components have balanced sizes result in a smaller $PWC_{\%}$ with respect to those that are not balanced.

3. Problem Formulation

We are interested in finding the minimum number of nodes (called *critical*), such that, their removal causes a performance degradation in the network in terms of $PWC_{\%}$. It is easy to note the intrinsic multi-objective nature of the problem: the two conflicting objectives are the minimization of the network connectivity and the minimization of the attack cost.

With the aim to provide an ILP formulation, we assume that the attacker is interested in finding a solution by:

1. maximizing the number of partitions;
2. minimizing the maximum partition cardinality;
3. minimizing the cost of the attack.

With reference to the Equation (2) and Figure 1, the maximization of partitions number (1), and, at the same time, the minimization of the largest partition size (2), correspond to the minimization of the PWC.

Let a connected undirected graph $G = \{V, E\}$; we assume that the attacker selects some nodes $v_j \in V_C \subseteq V$. Moreover, noting that, as a result of the attack, at most $n-1$ connected components can be obtained[1], in this regard we consider a set of $n-1$ *pairwise disjoint* partitions $V_1, \dots, V_{n-1}$ such that $V = V_C \cup_{i=1,\dots,n-1} V_i$.

We consider the Boolean variables $x_j^{(i)}$ for all $j = 1, \dots, n$ and for all $i = 1, \dots, n-1$, such that $x_j^{(i)} = 1$ when node v_j is assigned to the partition V_i, while we define c_j as a Boolean variable such that $c_j = 1$ if $v_j \in V_C$ and $c_j = 0$ otherwise.

With the aim to describe real scenarios, we introduce a vector $\mathbf{k}$ with n entries[2] $k_i \in [0, 1]$. Each entry k_i represents the cost associated to the removal of the i-th node.

The partitions $V_1, \dots, V_{n-1}$, as clarified later, reflect the way in which the nodes in the graph are separated after the attack. Specifically, in the following we prescribe that those nodes labelled as belonging to different partitions should not be connected by a path after the attacked nodes in V_C have been removed[3] in order to effectively characterize the attack.

In the proposed formulation, we allow some partition to be empty; the maximization of the number of non-empty partitions is one of the objectives of the attacker.

To model the ability of the attacker to select a given number of non-empty partitions, we introduce a Boolean variable t_i associated to each partition V_i. If V_i is empty then $t_i = 0$, otherwise $t_i = 1$.

Let us define the three objectives as follows.

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

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

[1] For instance, in a star graph if the central node is attacked, the network is disconnected in $n-1$ components, i.e., each isolated node.

[2] We reiterate that n is the number of nodes in the network.

[3] Notice that there is no guarantee that a partition contains a single connected component, i.e., each partition might be further decomposed in connected components.

2. Minimize the maximum partition cardinality:

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

3. Minimize the attack cost:

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

3.1. Problem Constraints

In order to provide an ILP formulation of the problem we introduce the following constraints.

In the first set of constraints (CS-I), we impose that each node has to be assigned[4] to only one set among $V_C, V_1, \ldots, V_{n-1}$:

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

In order to select decision variables that correspond to an attack that successfully disconnects the network in connected components, we impose that 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, in the second set of constraints (CS-II), which will be formally introduced later, we enforce that the following condition must hold:

$$(v_a, v_b) \notin E, \text{ for } v_a \in V_i, v_b \in V_j, \forall i, j = 1, \ldots, n-1. \tag{6}$$

Note that, any pair of partitions V_i and V_j (not considering V_C) and any pair of nodes v_a and v_b, must satisfy Equation (6). We can express the above-mentioned constraint as

$$A_{ab}\Big(x_a^{(i)} + x_b^{(j)}\Big) \leq 2 - \epsilon \tag{7}$$

[4] Recall that $\mathbf{c}$ and $\mathbf{x}^{(i)}$ are respectively the stack vectors of the variables c_i and $x_j^{(i)}$.

where $0 < \epsilon < 1$ is a coefficient required to avoid the use of a strict inequalities; the constraint is 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 constraint is violated when $v_a \in V_i$ and $v_b \in V_j$.

Let M be the incidence matrix of G and let M_+ and M_- be its non-negative and non-positive parts. We can write the aforementioned class of constraints (which we denote as *separation constraints*) in a compact form for a given pair of sets V_i and V_j and for all the edges as

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

Note that, in the proposed formulation, we consider two specular constraints for each edge, indeed we take into account undirected graphs and we have to explicitly handle (v_i, v_j) and (v_j, v_i) for all pairs of nodes connected by a link.

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

$$x_1^{(i)} + \ldots + x_n^{(i)} \geq t_i \qquad i = 1 \ldots n-1. \tag{CS-III}$$

According to the above constraint, a partition is not empty if it has at least one node assigned to itself.

We further simplify our problem, getting rid of the min and max in Equation (4) by introducing a new free variable $q \in \mathbb{N}$ and the additional constrains (CS-IV):

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

so that Equation (4) can be replaced by

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

As a result, the constraints in CS-IV enforce that

$$q \geq \max_{i=1,\dots,n-1} \left(\mathbf{1}_n^T \mathbf{x}^{(i)} \right) \tag{8}$$

and since q is minimized, the relation in Equation (8) is always satisfied as an equality.

The aim of the last set of constraints (CS-V), is to guarantee the presence of at least one critical node:

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

In fact, this inequality is satisfied if at least one node is not assigned to any of the $n-1$ partitions $V_1, \dots, V_{n-1}$. As a consequence, due to the presence of the constraints in CS-I, the “missing” node belongs to the critical set.

3.2. Objective Function

Let us now discuss the structure of the objective function; note that there are several alternative ways to define an objective function that considers the three objectives. In this paper, we choose to consider a convex combination of the three objectives, by introducing $\alpha_1, \alpha_2, \alpha_3$ such that

$$\alpha_i \in [0,1],\, i \in \{1,2,3\}, \quad \sum_{i=1}^{3} \alpha_i = 1. \tag{9}$$

in this way we are able to capture a large set of attackers’ behaviors, represented by different tradeoffs among the sub-objectives described in equations (3), (4), and (5).

As a result, the overall objective function becomes:

$$\min\left\{\underbrace{\frac{n}{\sum_{i=1}^{n} k_i}\alpha_1 \mathbf{k}^T\mathbf{c}}_{Attack \quad Cost} + \underbrace{\alpha_2 q - \alpha_3 \sum_{i=1}^{n-1} t_i}_{\widehat{PWC}}\right\}. \tag{10}$$

The first term of Equation (10) consists of the attack cost and it depends on the number of critical nodes assigned in $\mathbf{c}$ and the associated costs k_i. In addition to the parameter α_1, introduced to highlight the attacker preferences, we normalize the attack cost in order to avoid three unbalanced sub-objectives in Equation (10). The other terms aim to minimize an approximation of the PWC which is considerably easier to calculate than the standard PWC. We will provide more details about the relation between the PWC and the proposed approximation in Section 5.

In other words, in the proposed approach, we are replacing the PWC by the following approximation:

$$\widehat{PWC} = \gamma q - (1-\gamma)\sum_{i=1}^{n-1} t_i \tag{11}$$

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

Let us now provide a compact form that synthesizes the proposed ILP formulation; to this end, let

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

Moreover, we define the vector of independent variables as:

$$\mathbf{y} = \begin{bmatrix}\mathbf{c}^T, \mathbf{x}^T, q, \mathbf{t}^T\end{bmatrix}^T. \tag{12}$$

With the purpose of presenting the optimization problem by adopting the standard ILP form, we slightly modify constraints CS-I and CS-II.

Specifically, since the constraint CS-I is in equality form, in order to be represented in the standard ILP form we need to replace it by two inequality constraints:

$$\begin{aligned} -\mathbf{c} - \textstyle\sum_{i=1}^{n-1} \mathbf{x}^{(i)} \leq -\mathbf{1}_n, \\ \mathbf{c} + \textstyle\sum_{i=1}^{n-1} \mathbf{x}^{(i)} \leq \mathbf{1}_n. \end{aligned} \tag{13}$$

As for constraints CS-II, we consider a matrix

$$\mathcal{D} = \begin{bmatrix} M_+^{(n-1)} \otimes M_+ + M_-^{(n-1)} \otimes M_- \\ -M_+^{(n-1)} \otimes M_- - M_-^{(n-1)} \otimes M_+ \end{bmatrix}, \tag{14}$$

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, their Kronecker product $M_+^{(n-1)} \otimes M_+$ has $\frac{1}{2}\xi$ rows.

Therefore, with $\mathcal{D}$, we express constraints CS-II in a compact form as

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

At this point each set of constraints is represented in standard form and our optimization problem[5] is given by:

$$\begin{gathered} \min_{\mathbf{y}} \mathbf{r}^T \mathbf{y} \\ \text{Subject to} \\ \begin{cases} \mathcal{A}\mathbf{y} \leq \mathcal{B}; \\ \mathbf{y} \in \{0,1\}^{n+n^2} \cup \mathbb{N} \end{cases} \end{gathered} \tag{15}$$

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

[5] In the following, by $\mathbf{y} \in \{0,1\}^{n+n^2} \cup \mathbb{N}$ we mean that the only natural variable in $\mathbf{y}$ is q, while all other entries of $\mathbf{y}$ are Boolean.

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

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

In more details, the constraints in CS-I are collected in the first two rows of $\mathcal{A}$ and $\mathcal{B}$. The constrains described in CS-II are represented by the third row of the same matrices with reference to the matrix $\mathcal{D}$ defined in Equation (14). Finally, the constraints described in CS-III, CS-IV, and CS-V are represented by the last three rows of $\mathcal{A}$ and $\mathcal{B}$, respectively.

The entries 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. *By some algebra, it can be shown that matrix $\mathcal{A}$ has a number of rows*

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

where n is the number of nodes and e is the number of edges in G. Therefore, our ILP formulation (15) *has $O(n^2 e)$ constraints.*

As a consequence of the mentioned remark, we note that sparse graphs where $e \ll n^2$ must satisfy a reduced number of constraints, while denser graphs where $e \approx n^2$ need to satisfy $O(n^4)$ constraints.

4. Heuristic Algorithm

Instead of solving the ILP problem defined in 15 exactly, in this section we develop an heuristic approach that provides an approximated solution in reasonable time by sampling a large number of solutions and by selecting the feasible solution in terms of minimum cost, among the sampled ones.

It should be noted that, since problem (15) requires to specify $O(n^2)$ Boolean variables, and such variables have complex relations that must be verified (e.g., the separation constraints), it is hard to apply a simple brute-force Monte Carlo approach. In fact, there is the risk that a large fraction of the solutions thus generated are unfeasible.

However, we notice that q and $\mathbf{t}$ depend on the node partitioning process, i.e., we can easily find admissible choices for q and $\mathbf{t}$ given an admissible choice for the entries of $\mathbf{x}$.

Based on this intuition, the proposed *Feasible Solution Generation* (FSG) Algorithm (the pseudocode is reported in Algorithm 1 and, for space reasons, it continues in Algorithm 2) aims at providing random feasible solutions to be evaluated within the main heuristic algorithm.

Using the FSG Algorithm, we assign each node to a partition by considering its neighborhood. We notice, in fact, that if all already assigned neighbors of a node v_i belong to the same partition, then, a feasible solution is to assign also v_i to the same partition (e.g., see Figure 2a). Moreover, if a node v_i has neighbors assigned to different partitions, the only feasible choice is to set v_i as a critical node in order to preserve the absence of links between partitions (e.g., see Figure 2b).

Within the FSG Algorithm, we assume that only $m \leq n-1$ partitions can be nonempty, and we evaluate the nodes in V sequentially in random order. Specifically, we assign each node v_i to a partition (or to the set of critical nodes) as follows.

If no neighbor of v_i is assigned (lines $15-26$), we have two possible subcases. If there is an empty partition left (lines $16-21$), we randomly assign v_i to one of them (h) and we increase the number ϕ of the non-empty partitions by one. Moreover, for each neighbor v_j in $\mathcal{N}_i$, the list of assigned neighbors, $\mathcal{M}_j$, is updated. Otherwise (lines $23-24$), we reconsider the node again by putting it back in the set of not yet considered nodes. We perform this procedure a

Algorithm 1: Feasible Solution Generator (continues on Algorithm 2)

```
Data: Graph $G = \{V, E\}$, maximum number $m$ of partitions, maximum
      number $\chi_{\max}$ of reconsiderations of selected nodes
Result: Feasible solution $\mathbf{y}$
$c_j \leftarrow 0$, $\forall j = 1, \ldots n$;
$x_j^{(i)} \leftarrow 0$, $\forall i = 1, \ldots, m$ and $\forall j = 1, \ldots n$;
$t_i \leftarrow 0$, $\forall i = 1, \ldots, m$;
$q \leftarrow 0$;
$\phi \leftarrow 0$; /* nonempty partitions */
$\chi \leftarrow 0$; /* no node has been reconsidered */
for $i = 1 \ldots n$ do
    $\mathcal{M}_i \leftarrow \emptyset$; /* assigned neighbors of $v_i$ */
    $p(i) \leftarrow 0$; /* $v_i$ is not assigned */
end
$\mathcal{I} \leftarrow V$;
while $\mathcal{I} \neq \emptyset$ and $\chi < \chi_{max}$ do
    select random $v_i \in \mathcal{I}$;
    $\mathcal{I} \leftarrow \mathcal{I} \backslash \{v_i\}$;
    /* If no neighbor of $v_i$ is assigned add $v_i$ to a random partition
       or reconsider it later */
    if $\mathcal{M}_i = \emptyset$ then
        /* If there is an empty partition, attempt to assign $v_i$ to a
           random empty partition */
        if $\phi < m$ then
            select random $h \in \{1, \ldots, m\}$ such that $V_h = \emptyset$;
            $p(i) \leftarrow h$;
            $\phi \leftarrow \phi + 1$;
            $\mathcal{M}_j \leftarrow \mathcal{M}_j \cup \{v_i\}$ for all $v_j \in \mathcal{N}_i$;
        end
        /* Reconsider the node $v_i$ later to avoid too many critical
           nodes */
        else
            $\mathcal{I} \leftarrow \mathcal{I} \cup \{v_i\}$; /* put back $v_i$ in $\mathcal{I}$ */
            $\chi \leftarrow \chi + 1$; /* increase reconsiderations */
        end
    end
    else
        /* If all already assigned neighbors of $v_i$ are in the same
           partition $h$, assign $v_i$ to partition $h$ */
        if $\forall v_j \in \mathcal{M}_i$, $p(j) = h$ for some $h > 0$ then
            $p(i) \leftarrow h$;
            $\mathcal{M}_j \leftarrow \mathcal{M}_j \cup \{v_i\}$ for all $v_j \in \mathcal{N}_i$;
        end
        /* Assign $v_i$ to the set of critical nodes */
        else
            /* Do nothing, critical nodes are assigned at the end */
        end
    end
end
```

Algorithm 2: Feasible Solution Generator (continuation)

```
   /* Choose assignment variables */
36 for $i = 1 \dots n$ do
37     $x_i^{(p_i)} \leftarrow 1$;
38 end
   /* Assign Critical Nodes */
39 for $j = 1 \dots n$ do
40     $c_j \leftarrow 1 - \sum_{i=1}^{m} x_j^{(i)}$;
41 end
   /* select $t_i$ */
42 for $i = 1 \dots m$ do
43     if $|x^{(i)}| > 0$ then
44         $t_i \leftarrow 1$;
45     end
46 end
47 $q \leftarrow \max\{|x^{(1)}|, \dots, |x^{(m)}|\}$;
48 Return: $y = [c_1, \dots, c_n, x_1^{(1)}, \dots, x_n^{(n-1)}, q, t_1, \dots, t_{n-1}]^T$
```

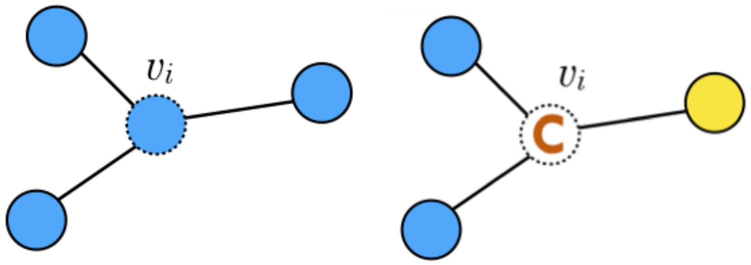
maximum of $\chi_{\max}$ times.

If, conversely, at least a neighbor of v_i belongs to a partition (lines $27 - 34$), we have two possible sub-cases. If all the already assigned neighbors of v_i belong to the same partition h (lines $28 - -31$), we assign v_i to the same partition h. Moreover, for each neighbor v_j of v_i, the list of assigned neighbors, $\mathcal{M}_j$, is updated. If none of the above cases is verified (lines $32 - 33$), then v_i has neighbors assigned to different partitions and it is labeled as critical.

We let the procedure end when all the nodes have been assigned or when a maximum χ_{max} of reconsiderations has been done. In the latter case notice that each not assigned node is labeled as critical. Note further that we actively label critical nodes only at the end of the main cycle, using constraint (CS-I) once all $x_j^{(i)}$ have been specified.

We point out that, with the aim to avoid feasible solutions with a large number of critical nodes, an appropriate choice for the parameter χ_{max} is at least one order of magnitude greater than n.

We conclude the algorithm by calculating the entries of $\mathbf{y}$ and the value of q based on the partitions assignments. Note that, in addition to ensure feasible solutions, the algorithm also guarantees the internal connectivity for each partition.[6]



(a) All the neighbors of v_i are assigned to the same partition: a feasible solution is to assign also v_i to the same partition.

(b) The neighbors of v_i are assigned to different partitions: a feasible solution is to label v_i as critical.

Figure 2: Heuristic assignment criterion adopted within Algorithm 1.

Remark 2. *Algorithm 1 always provides a feasible solution. In fact, it should be noted that, by assigning a node to the set of critical nodes whenever its neighbors belong to more than one partition ensures that the separation constraints are enforced.*

On the basis of the aforementioned algorithm, the heuristic approach adopted in this paper is summarized in Algorithm 3, where a large number n_a of solutions is created via the FSG Algorithm, with a random integer number m_{tmp} of allowed partitions. Note that, in some cases (e.g., for large networks), the optimal solution is unlikely to contain a number of partitions that is $O(n)$; therefore, a possible choice is to arbitrarily fix a maximum number of partitions $m_{\max} \ll n$ within Algorithm 3. In the proposed algorithm, a value $m_{tmp} \in \{2, \dots, m_{\max}\}$ is selected at each round with a probability

$$Pr(m_{tmp} = q) = \frac{\frac{1}{q}}{\sum_{h=2}^{m_{\max}} \frac{1}{h}} = \frac{\prod_{h=2}^{m_{\max}} h}{q \sum_{h=2}^{m_{\max}} \prod_{l=2, l \neq h}^{m_{\max}} h}, \tag{16}$$

for all $q \in \{2, \dots, m_{\max}\}$. This choice is made so that it is more likely to

[6] Recall that, in our formulation there is, in general, no guarantee that the partitions are internally connected.

generate instances with a limited number of partitions.

Consequently, the algorithm evaluates the solutions in terms of their objective function, selecting the best one.

Algorithm 3: Heuristic approach for network vulnerabilities detection.

```
Data: Graph $G = \{V, E\}$
Number of attempts $n_a$
Cost vector $r$
maximum number of partitions $m_{\max} \leq n$
Result: Approximated solution $y_{\max}$
$y_{\max} \leftarrow \emptyset$;
$z_{\max} \leftarrow +\infty$;
for $i = 1, \dots, n_a$ do
    select $m_{tmp} \in \{2, \dots, m_{\max}\}$ with a probability as given in
    Equation (16);
    $y_{tmp} \leftarrow FSG(G, m_{tmp})$;
    if $r^T y_{tmp} < z_{\max}$ then
        $z_{\max} \leftarrow r^T y_{tmp}$;
        $y_{\max} \leftarrow y_{tmp}$;
    end
end
Return: $y_{\max}$
```

5. Simulation Results

In this section we start by showing the correlation between the PWC as defined in Equation (1) and the linear approximation $\widehat{PWC}$ introduced in Equation (11), in order to justify the adoption of the latter metric. Then, we illustrate the effectiveness of the proposed ILP optimization method on a particular network. Finally, we apply the heuristic approach described in Algorithm 3 on a real network composed of 332 nodes and 2126 links representing the routes among the US airports in 1997.

5.1. PWC Approximation

In this subsection we provide an experimental validation of our approximated PWC index, namely $\widehat{PWC}$ defined in Equation (11). Recall that such an index is a linear combination of two of the three terms that constitute the objective function of the proposed formulation, i.e., the number of connected components and the size of the largest connected component.

The proposed validation strategy consists in analyzing the correlation between PWC and $\widehat{PWC}$ over an instance of a graph with $n = 40$ nodes. Specifically, we sample 3800 admissible solutions for a specific value of the $\widehat{PWC}$, each one corresponding to a PWC value depending on the choice of the parameter α, which represents the trade-off between the two sub-objectives.

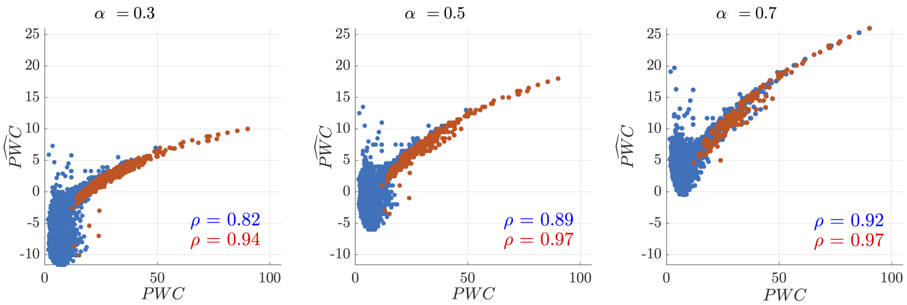


Figure 3: Correlation between PWC and $\widehat{PWC}$ for three different values of the parameter α, considering 3800 feasible instances. The correlation values shown in red are related only to the subset of red points (i.e., those corresponding to solutions where at most 6 nodes are attacked), while the correlation values shown in blue are related to all the sets of points (i.e., the red and the blue points).

In each of the three panels in Figure 3 we select a specific value for α and we calculate the PWC and the $\widehat{PWC}$ associated to each sampled solution. Specifically, in order to consider realistic situations where just few nodes are attacked, we show in red the subset of solutions corresponding to those attacks targeting up to 6 nodes (e.g., 15% of all nodes); moreover, we consider the whole set of sampled solutions, which includes both the red and the blue dots (i.e., those featuring more than 6 attacked nodes).

As shown by Figure 3, PWC and $\widehat{PWC}$ appear tightly correlated (the correlation is always larger than or equal to $\rho = 0.82$). Particularly, the correlation is larger in the case of a limited number of attacked nodes (e.g., the red cloud of points in the figure). This is highly beneficial, since we are interested in finding cheap solution in terms of cost of attack, as these solutions tend to be more representative of real attack strategies.

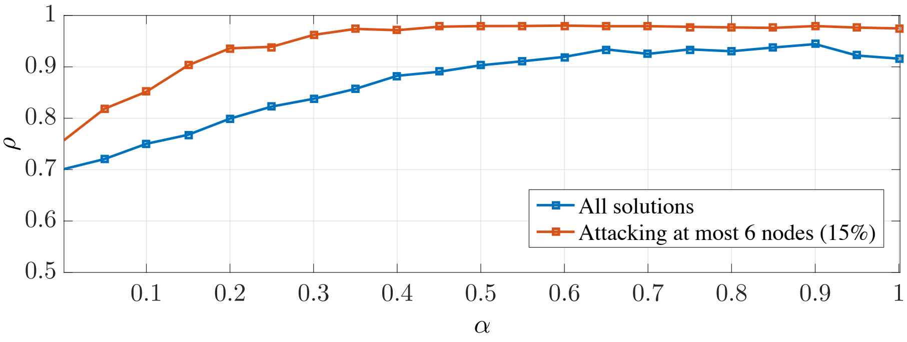


Figure 4: Correlation between PWC and $\widehat{PWC}$ for 21 different values of the parameter α.

To further corroborate the ability of the proposed index $\widehat{PWC}$ to closely approximate the PWC, we show, in Figure 4, the relation between the parameter α, which determines the trade-off between the two sub-objectives that constitute the $\widehat{PWC}$, and the correlation coefficient ρ between the PWC and the $\widehat{PWC}$. For each considered choice of α, we generate 2000 feasible solutions over the same graph with $n = 40$ nodes. Also in this case we report in blue the correlations based on all the solutions while, in order to consider realistic situations where just few nodes are attacked, we show in red the correlation associated to the subset of solutions corresponding to the attacks targeting up to 15% of all nodes. By analysing the set of solutions, the results show that the correlation, already high for small α values, tends to grow with α, by reaching a plateau of 0.98 (red line) and 0.90 (blue line) for $\alpha \geq 0.5$.

5.2. ILP Formulation

This subsection purpose is to show the potential of the proposed ILP formulation; specifically, we analyze the optimal solution to our formulation over a sample instance, for different trade-offs among the three objectives that constitute the objective function of our formulation.

Let us consider the graph in Figure 5 with $n = 25$ nodes and let us calculate

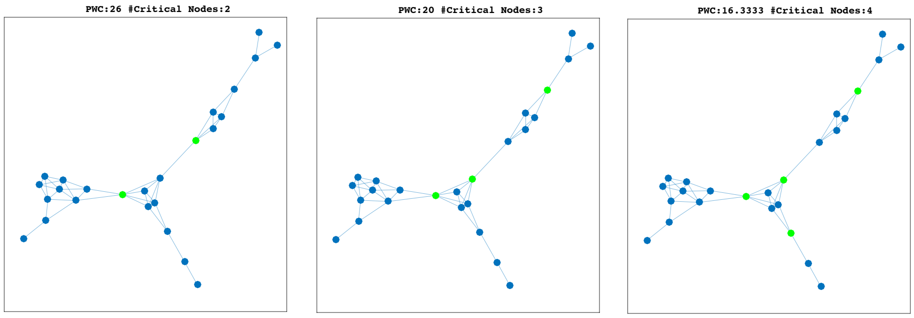


Figure 5: Three optimal solutions (for particular choices of the parameters within the objective function of our ILP formulation) associated to a low number of removed nodes (green) and a low rate of PWC. In the left panel, the network has been partitioned in three partitions by removing two nodes (PWC : 26%); in the central panel the network is broken down into 4 partitions attacking three nodes (PWC : 20%); on the right panel, four nodes are attacked and the network is decomposed in five partitions (PWC : 16.3%).

the optimal solution to our formulation for 66 different combinations of the parameters α_1, α_2, and α_3 considering an attack cost $k_i = 1$ for each node.

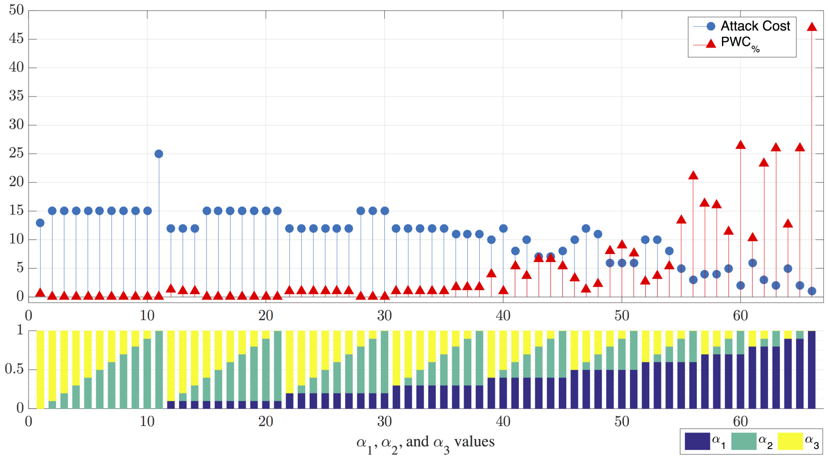


Figure 6: $PWC_{\%}$ and number of critical nodes found within the optimal solutions of our ILP formulation, considering the graph in Figure 5. The solutions are computed for 66 different combinations of α_1, α_2, and α_3.

In Figure 6 we show the $PWC_{\%}$ (with a red line) and the number of critical nodes (in blue) for each triple of parameters, while the corresponding combination of the parameters is shown via stacked plot on the x-axis. According to

Figure 6, as long as the value of α_1 (i.e., the weight of the objective term related to minimize the attack cost that in the example matches with the number of attacked nodes) is less than 0.4, the optimal solution involves an high number of critical nodes and, as a result, a value of $PWC_{\%}$ near zero is obtained. Therefore, these solutions hardly describe the behavior of a real attacker. Starting from the solutions associated to $\alpha_1 \geq 0.4$, the number of attacked nodes slowly decreases, while the $PWC_{\%}$ increases. Among other combinations, Figure 5 shows the results of three particular choices of weights, which correspond to the best solutions associated to the removal of 2, 3, and 4 nodes, respectively (in Figure 5, for each of the three above cases, we show in green the attacked nodes). The results in terms of $PWC_{\%}$ and attacked nodes are shown in Table 1. It is worth noting that the resulting $PWC_{\%}$ in the three cases is quite similar (26%, 20%, and $16,3\%$, respectively), although the parameters that yield this result in the three cases are considerably different. Conversely, the proposed framework seems to be much more descriptive of the subtle trade-offs among the different clashing objectives for the attacker.

Table 1: Optimal solutions details.

α_1	α_2	α_3	PWC	$\lvert V_c \rvert$
0.9	0.1	0	26%	2
0.6	0.4	0	20%	3
0.7	0	0.3	16.3%	4

5.3. Case Study: US Flight Tracks

In this subsection we use the heuristic approach described in Section 4, with the aim to find good solutions to our ILP formulation in reasonable time, and we will compare these results with another attack strategy already known in literature.

To do this we analyze the American airport network USAir97 [21] as it was in 1997; as show in Figure 7 this network is composed of 332 nodes and 2126 edges, as shown in Figure. Each node represents an airport, while each edge is associated to a direct flight from an airport to another. For the sake of clarity, in Figure 8, we show the degrees distribution of USAir97, i.e., the number of flight routes for each airport. Most of the nodes are weakly connected to the others but there is a small subset of nodes with an high degree (the hubs).

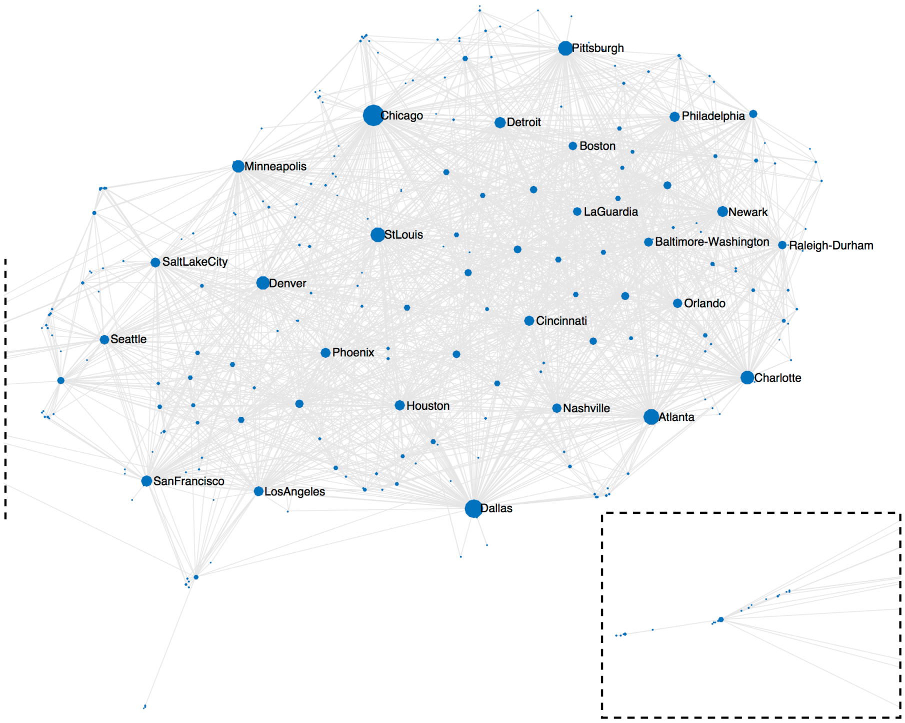


Figure 7: The USAir97 network [21] (for space reasons, a portion of the network is shown in the box in the lower-right corner). The node size is proportional to its degree.

Clearly, in a real scenario, major airports are more protected than small airports. To model this fact, we assume that the attack cost is proportional to the relevance of the airport. In other words, we assume that the cost k_i of an attack against the i-th node is equal to its degree. This means that the attack cost of the Chicago Airport (the largest hub) is 139 while the cost associated to the Abilene Regional Airport is 1 because it is only connected to the airport of Dallas. With reference to Algorithm 3, we consider $n_a = 8000$ attempts and we set the attack costs k_i equal to the node; moreover, the maximum number of partitions is set to 4, and we choose $\chi_{max} = 3000$. The objective function has been evaluated for n_a solutions, considering 10 different combinations of values for the weights α_1, α_2, and α_3. In this way, we reproduce ten different patterns of attack.

With the aim to analyze the effectiveness of the proposed attack strategy, we compare our solutions with another attack strategy which is often adopted in literature. Specifically, we consider the strategy of iteratively disconnecting the nodes of the network by descending order of degree, since this approach has been shown to be highly disruptive for the network connectivity (see, among others [2, 22]).

In Table 2 and Figure 9 we show the results of the heuristic approach and we compare them with the *degree-based* strategy. The left side of Table 2 collects the results of the heuristic approach in terms of cost (i.e. spent budget), objective function weights $(\alpha_1, \alpha_2, \alpha_3)$, number of attacked nodes and their IDs, and $PWC_{\%}$ value. Despite they derive from different triple of weights, we observe that different attack strategies converge to the same targets and so to the same costs. Moreover, we note that, if the value of α_1 decreases, the attack cost and the number of attacked nodes increases, and consequently the $PWC_{\%}$ decreases. On the right side of the table we report the results of the degree-based attack strategy with the same budgets.

Specifically, we assume that the attacker has a fixed budget, which corresponds to the cost of the solution found via our heuristic approach, then we apply the degree-based attack strategy, which iteratively selects the nodes with the largest degree, until there is budget left. Note that the budget and the attack cost are normalized in the range $[0, \ldots, 332]$, in order to be comparable with the cost of the attack in Equation (10). It is noteworthy that with a fixed budget and by attacking several small nodes rather few hubs, a large degradation in terms of PWC can be achieved. Moreover, the selection of target nodes on the base of their relevance (i.e. their degree) produces a limited degradation in terms of PWC. Conversely, by applying our algorithm we are able to considerably degradate the PWC. This is achieved by focusing the limited resources of the attacker on a set of small airports, rather than targeting huge hubs. The results in terms of $PWC_{\%}$ (5^{th} and 8^{th} columns) highlight the effectiveness of

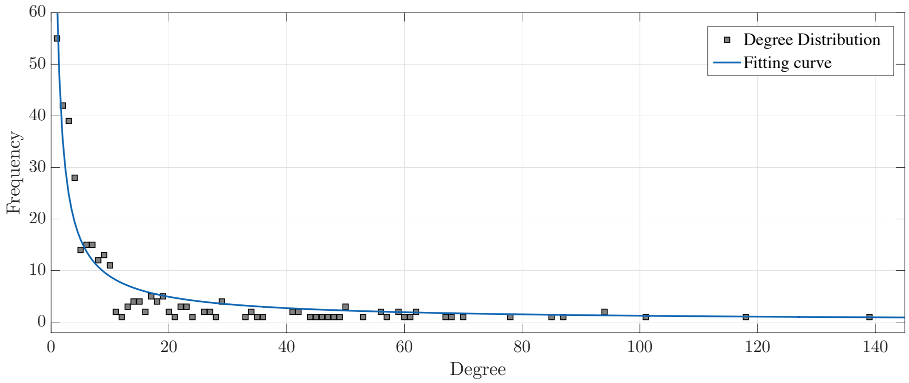


Figure 8: USAir97 degree distribution. The markers represent the frequency distribution of degree and the blue line denotes the fitting curve.

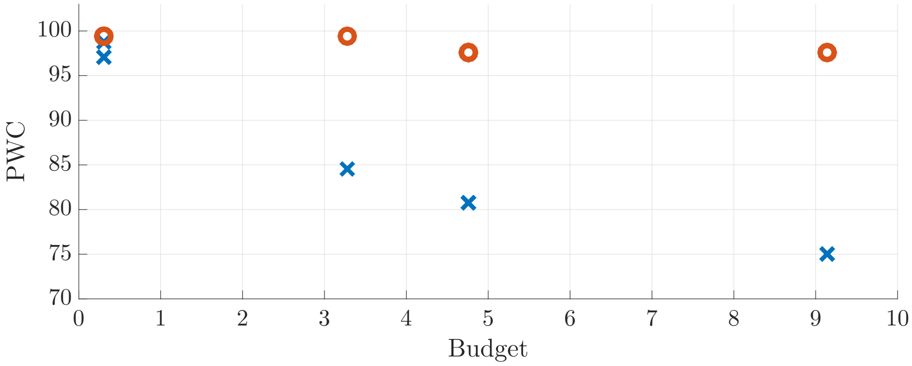


Figure 9: Results in terms of $PWC_{\%}$ and budgets of the application of the heuristic approach (blue cross) and the degree strategy (red circle).

the proposed approach; specifically, the particular strategy identified by our algorithm causes the disconnection of the network into several partitions; this may potentially cause a large damage to the connectivity of the network.

6. Conclusion

In this paper we formulate an optimization problem in order to find critical nodes, i.e., nodes whose removal has severe effects on the connectivity of the network. In particular, we adopt an attacker perspective and we assume that he/she has the conflicting objectives of minimizing the ability of the nodes to

Table 2: Analysis of the heuristic approach for 10 combinations of parameter $\alpha_1, \alpha_2, \alpha_3$, and comparison in terms of $PWC_{\%}$ with the degree strategy.

	Heuristic Approach				Degree-Based Attack Strategy		
Budget	Alpha Values	# Attacked Nodes	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**
	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**

communicate with each other while keeping the cost of the attack at a minimum. Unlike previous literature, we do not make any assumption about the number of partitions that are obtained after the attack or about the number of attacked nodes. In the proposed formulation we consider the trade-offs among the clashing objectives of the attacker, each mediated by specific weights that model the preferences of the attackers.

Modeling the attacker behavior and identifying critical sites is the first step in order to raise network protection. Future work will be mainly devoted to consider situations in which the attacker pays different, possibly dynamically changing costs for attacking different sites, and to provide decision support systems to help the decision makers identify sites to be protected. We will also consider to complement the framework by introducing an optimization problem for the defendant and we will inspect the possibility to cast these coupled optimization problems in the framework of the game theory, seeking those solutions that may lead to an equilibrium.

References

[1] R. Albert, H. Jeong and A. L. Barabási, Error and attack tolerance of complex networks, *Nature*, vol. 406(6794), pp. 378–382, 2000.

[2] P. Holme, B. J. Kim, C. N. Yoon and S. K. Han, Attack vulnerability of complex networks. *Physical Review E*, 65(5), 056109, 2002.

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

[4] X. Huang, J. Gao, S. V. Buldyrev, S. Havlin and H. E. Stanley, Robustness of interdependent networks under targeted attack, *Physical Review E*, vol. 83(6), 065101, 2011.

[5] S. Shao, X. Huang, H. E. Stanley and S. Havlin, Percolation of localized attack on complex networks, *New Journal of Physics*, vol. 17(2), 02304, 2015

[6] V. H. Louzada, F. Daolio, H. J. 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. Gabryś (Eds.), Springer International Publishing, Zurich, Switzerland, pp. 215–225, 2015.

[7] Y. Berezin, A. Bashan, M. M. Danziger, D. Li and S. & Havlin, Localized attacks on spatially embedded networks with dependencies, *Scientific reports*, vol. 5, 8934, 2015.

[8] European Commission, The Directive on security of network and information systems (NIS Directive), Strasbourg, France, 2016.

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

[10] 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.

[11] 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.

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

[13] 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.

[14] A. Arulselvan, C. W. Commander, O. Shylo and P. M. Pardalos, Cardinality-constrained critical node detection problem. In *Performance models and risk management in communications systems*, N. Gulpinar, P. Harrison and B. R ustem (Eds.), Springer, New York, pp. 79–91, 2011.

[15] Y. Shen, N. P. Nguyen, Y. Xuan and M. T. 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.

[16] 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*, vol. 20(2), pp. 609–619, 2012.

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

[18] M. Ventresca, K. R. Harrison and B. M. Ombuki-Berman, An Experimental Evaluation of Multi-objective Evolutionary Algorithms for Detecting Critical Nodes in Complex Networks, *Proceedings of the European Conference on the Applications of Evolutionary Computation*, pp. 164-176, 2015.

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

[20] L. Faramondi, G. Oliva, F. Pascucci, S. Panzieri and R. Setola, Critical Node Detection based on Attacker Preferences, *Proceedings of the 24th Mediterranean Conference on Control and Automation*, pp. 773–778, 2016.

[21] Batagelj V, Mrvar A. Pajek datasets; 2006.

[22] Z. M. Lu and X. F. Li, Attack Vulnerability of Network Controllability, *PloS one*, vol. 11(9), e0162289, 2016.

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

[24] G. G. Brown, W. M. Carlyle, J. Salmerón and K. Wood, Analyzing the vulnerability of critical infrastructure to attack and planning defenses, *Proceedings of the INFORMS Annual Meeting*, pp. 102–123, 2005.

[25] T. G. Lewis, *Critical infrastructure protection in homeland security: defending a networked nation*, John Wiley & Sons, New York, United States of America, 2014.

[26] O. Lordan, J. M. 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.

[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.

[28] 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.