ORIGINAL ARTICLE

Networks

Min-Max Optimization of Node-targeted Attacks in Service Networks

Bernard Fortz[1,2,3] | Mariusz Mycek[4] | Michał Pióro[4,5] | Artur Tomaszewski[4]

[1] HEC - Management School of the University of Liège, Liège, Belgium

[2] Department of Computer Science, Université Libre de Bruxelles, Brussels, Belgium

[3] INOCS, INRIA, Lille, France

[4] Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland

[5] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland

Correspondence
Bernard Fortz, HEC - Management School of the University of Liège, Rue Louvrex, 14, 4000 Liège, Belgium
Email: bernard.fortz@uliege.be

Funding information
POB Research Centre Cybersecurity and Data Science of Warsaw University of Technology within the Excellence Initiative Program – Research University (ID-UB), Poland.

This paper considers resilience of service networks that are composed of service and control nodes to node-targeted attacks. Two complementary problems of selecting attacked nodes and placing control nodes reflect the interaction between the network operator and the network attacker. This interaction can be analyzed within the framework of game theory. Considering the limited performance of the previously introduced iterative solution algorithms based on non-compact problem models, new compact integer programming formulations of the node attack optimization problem are proposed, which are based on the notion of pseudo-components and on a bilevel model. The efficiency of the new formulations is illustrated by the numerical study that uses two reference networks (medium-size and large-size), and a wide range of the sizes of attacks and controllers placements.

KEYWORDS
service networks, SDN, control nodes, node attacks, controllers placement, resilience, optimization and integer programming

1 | INTRODUCTION

We consider a network that offers some kind of service in a given set of network locations. Each location houses a service node that actually provides the service, and might also house a control node (controller in short). The service node needs a controller to operate: for that it may use either the local controller that is placed in the same location or, if the location does not house a controller, a remote controller in some other location. In the latter case the service node must communicate with the controller using a sequence of links that interconnect the locations.

In the ICT domain the considered network model directly reflects a number of computer networking scenarios. In particular, in software defined networks (SDN) [14, 7] a limited set of SDN controllers controls the operations of a large number of data switches. And in content delivery networks (CDN) the nodes that deliver content to customers (there are many such nodes as they must be placed relatively close to the customers) download the content from a limited number of nodes where the content is stored (one may notice that those storage nodes, which actually store copies of the original data, must in turn communicate with primary sources of content, i.e., the nodes where content originates). However, one may find applications of the considered model in other domains as well, would it be utilities, manufacturing, logistics, sales, or medicine. In those domains the service nodes and the control nodes might correspond, respectively, to power dispatch stations and power plants, factories and transportation hubs, dispatch centers or warehouses and factories, sale points or supermarkets and warehouses, clinics or testing points and medical laboratories, etc. Depending on the context, those nodes are interconnected with different kinds of communications, transportation and utility networks, and their inaccessibility may result not only from attacks, but, e.g., from technical malfunctioning and natural disasters.

We aim at protecting the considered network of service and control nodes against targeted node attacks, assuming that the attack targets a set of selected network locations making both service nodes and control nodes (if any) at those locations unavailable. Moreover, the attack makes unavailable the network links that are terminated at the attacked locations, potentially disconnecting the network graph into a number of (connected) components. After the attack, the service node still provides service (and is called a surviving (service) node) only if its location has not been attacked and the network component it belongs to still contains at least one location with a control node.

In our previous (conference) paper [33] we introduced two complementary optimization problems of protecting the network of the considered class against targeted node attacks. The first problem of optimizing the controllers placement consists in selecting a set of locations to place controllers so as to maximize the number of service nodes that survive the attack no matter what set (of a given size) of locations is attacked. The complementary problem of optimizing the node-targeted attack, in turn, consists in finding a set (also of a given size) of locations to attack so as to minimize the number of service nodes that survive the attack no matter where the controllers are placed. While the first problem is a problem the network operator is faced with, the second problem is a problem of the network attacker. Arguably, those two problems of placing control nodes and selecting attacked nodes reflect the interaction between the network operator and the network attacker, which can be analyzed within the framework of game theory.

To solve the two problems, in the above mentioned conference paper we proposed two symmetrical non-compact integer programming (IP) problem formulations, one for optimizing the controllers placements, and another for optimizing the node attacks. The formulations are based on the lists of, respectively, worst node attacks and best controllers placements. We then developed two iterative solution algorithms that generate either worst attacks or best placements by solving the two models alternately. We note that while related issues had been widely investigated in the literature (mainly in the context of SDN), the problem formulations and their solution algorithms proposed in our paper were original.

In [33] we tested the proposed problem formulations and solution algorithms on two networks (one medium-size and one large-size), specified in the SNDlib library [21], using a sample pair of attack and placement sizes. The experiments showed limited computational efficiency of optimization algorithms for the large-size network. Especially visible were the limitations of the attack optimization algorithm, with which we were unable to solve to optimality most of the problem instances for that network.

With this motivation, in the current paper we concentrate on the attack optimization problem. We propose two computationally efficient compact IP problem formulations utilizing an approach based on a notion of pseudo-components, and on an approach based on bilevel optimization. We also run extensive numerical experiments, testing the performance of the formulations using the same medium- and large-size networks as before, but for a wide range of attack and placement sizes.

The composition of the paper is the following. We start with Section 2 containing a short survey of the related work, and then, in Section 3, we introduce the notation that we use throughout the paper and describe the two complementary optimization problems, for which we present a small problem example to illustrate their solutions. In Section 4 we define a non-compact model of the attack optimization problem. Next, in Section 5, which contains the main theoretical contribution of the paper, we systematically develop a set of compact IP formulations of the problem, and then, in Section 6, we discuss selected issues related to efficiency of the formulations and their extensions. Section 7 contains a description and results of a detailed study of the computational performance of the proposed formulations. Finally, in Section 8 we conclude the paper.

2 | RELATED WORK

The two topics discussed in this article, i.e., optimizing controller deployment and planning attacks against controller-equipped network nodes, are widely discussed in the literature, mainly in the context of SDN.

Since controller placement optimization is not the main concern of this paper, we only mention here that although this topic is dealt with in numerous surveys (for example, [32, 18, 7]) and numerous research papers (for example, [13, 22, 25]), most of the work described there does not consider resilience to attacks, and is devoted to optimizing controllers placements with respect to such measures as controller-to-controller delay and (service) node-to-controller delay. Controller placement problems directly related to the one formulated in this paper (CPOP, Section 3.2) are considered in such papers as [28, 27, 29], and also in more recent papers [3, 20, 24]. However, all these papers assume fixed lists of attacks, pre-calculated without taking into account possible locations of controllers, while in this paper, and in [33], the problem is to find a placement of controllers that maximizes the number of surviving service nodes, no matter what set (of a given size) of node locations is attacked. An exception is paper [23], in which the lists of attacks used are constructed assuming a probability distribution of controller locations considered by the network operator.

Now let us proceed to work related to the main problem considered in this paper (NAOP, Section 3.2), which is to find an attack of a given size that minimizes the number of surviving service nodes, no matter where the controllers are placed. An important observation is that virtually all studies of planning attacks on nodes reported in the literature do not consider possible controllers placements at all, but only use network graph topology connectivity measures. The papers of this kind are numerous, see for example [26, 3] and the references therein. The most advanced (and effective) of such topological attacks are obtained by solving the so-called Critical Node Detection (CND) problem, which finds a set of nodes (of a given cardinality) whose removal will, roughly speaking, split the network graph into as many mutually disjoint components as possible. Exact methods to solve this problem can be found in [1, 30, 17]. Some of those methods can be quite efficient in practice but since the CDN problem is $\mathcal{NP}$-hard, heuristic methods of finding critical nodes have also been proposed [11, 8], including very simple heuristics, such as deleting the nodes of the highest degree [10, 19, 15, 2]. To our knowledge, the only exception is again paper [23], which takes into account, in a probabilistic way, potential controllers locations when optimizing attacks.

3 | NOTATION AND PROBLEM DESCRIPTION

3.1 | Notation

We consider a service network represented by a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of vertices $\mathcal{V} = \{1, 2, \ldots, V\}$ represents network locations, each containing a service node (node in short). The set of undirected edges $\mathcal{E} \subseteq \mathcal{V}^{|2|}$ (where $\mathcal{V}^{|2|}$ is the family of all 2-element subsets of the set of vertices $\mathcal{V}$) represents transport links (links in short). For each link $e \in \mathcal{E}$, its end nodes are denoted by $\alpha(e), \beta(e) \in \mathcal{V}$, and in the cases when node numbering is important we assume (w.l.o.g.) that $\alpha(e) < \beta(e)$. For each node $v \in \mathcal{V}$, the set of links incident with v is denoted by $\delta(v) = \{e \in \mathcal{E} : v \in \{\alpha(e), \beta(e)\}\}$, and $\deg(v) = |\delta(v)|$ denotes the degree of node v. Similarly, the set of nodes adjacent to node v is denoted by $\Delta(v) = \{w \in \mathcal{V} \setminus \{v\} : \{v, w\} \in \delta(v)\}$.

The network is equipped with controllers and $\mathcal{S}$ denotes the set of allowable controllers placements. A placement $s \in \mathcal{S}$ is characterized by the set of network locations $\mathcal{V}(s) \subseteq \mathcal{V}$ where controllers are placed (apart from the service nodes). A typical example of such a set $\mathcal{S}$ is the set of all M-node controllers placements (denoted by $\mathcal{S}[M]$ for a given integer parameter M, where $0 < M \leq V$), i.e., the set of all placements s with $V(s) := |\mathcal{V}(s)| = M$.

Possible attacks targeting network nodes are included in a given set $\mathcal{A}$. Each attack $a \in \mathcal{A}$ is characterized by the set of the attacked locations $\mathcal{V}(a)$ ($\mathcal{V}(a) \subseteq \mathcal{V}$). A typical example of set $\mathcal{A}$ is the set of all K-node attacks (denoted by $\mathcal{A}[K]$ for a given integer parameter K, where $0 < K < V$), i.e., the set of all attacks a with $V(a) := |\mathcal{V}(a)| = K$. Hence, the network graph $\mathcal{G}$ affected by attack a is reduced to subgraph $\mathcal{G}(a) = (\mathcal{V} \setminus \mathcal{V}(a), \mathcal{E} \setminus \bigcup_{v \in \mathcal{V}(a)} \delta(v))$, and the family $\mathcal{C}(a)$ connected components into which graph G is split as the result of attack a (induced by attack a) is the set of $C(a) := |\mathcal{C}(a)|$ maximal connected subgraphs of $\mathcal{G}(a)$. For each component $c \in \mathcal{C}(a)$, $\mathcal{V}(c)$ denotes the set of its $V(c) := |\mathcal{V}(c)|$ nodes. In the following, when it does not lead to a misunderstanding, the term "component" will also be used for the set of component's nodes.

Finally, we assume that as a result of an attack a, at each directly attacked location in $\mathcal{V}(a)$ its service node and controller (if any) become out of service. Moreover, all service nodes in those components in $\mathcal{C}(a)$ that do not contain any controller also stop working. In effect, the service nodes that are still operational after the attack (called the *surviving nodes*) are precisely those nodes that belong to the *surviving components*, i.e., the components in $\mathcal{C}(a)$ that contain at least one controller. For a given placement $s \in \mathcal{S}$, such a subfamily is denoted $\mathcal{C}(s, a)$ and formally defined as follows:

$$\mathcal{C}(s, a) = \{c \in \mathcal{C}(a) : \mathcal{V}(c) \cap \mathcal{V}(s) \neq \emptyset\}. \tag{1}$$

The family of the remaining components, i.e., $\mathcal{C}(a) \setminus \mathcal{C}(s, a)$, is denoted by $\overline{\mathcal{C}}(s, a)$. Clearly,

$$\overline{\mathcal{C}}(s, a) = \{c \in \mathcal{C}(a) : \mathcal{V}(c) \cap \mathcal{V}(s) = \emptyset\}. \tag{2}$$

Note that for a given placement s and a given attack a the sets $\mathcal{V}(a)$, $\mathcal{C}(s, a)$ and $\overline{\mathcal{C}}(s, a)$ define a partition of the set of nodes:

$$\mathcal{V} = \mathcal{V}(a) \cup \bigcup_{c \in \mathcal{C}(s,a)} \mathcal{V}(c) \cup \bigcup_{c \in \overline{\mathcal{C}}(s,a)} \mathcal{V}(c). \tag{3}$$

The basic *resilience (to attack) measure* considered in this paper is the number of nodes, denoted by $V(s, a)$ ($s \in \mathcal{S}, a \in \mathcal{A}$), that survive a given attack a in the network equipped with controllers deployed according to placement s.

More precisely,

$$V(s,a) = \sum_{c \in C(s,a)} V(c), \qquad\qquad s \in \mathcal{S}, a \in \mathcal{A}. \tag{4}$$

Given this measure, we introduce the notions of the worst attack and the best controllers placement. Taking the operator's point of view, the *worst attack* with respect to a given placement $s \in \mathcal{S}$, denoted by $a(s)$, is defined as any attack a in $\mathcal{A}$ that minimizes the number of surviving nodes $V(s,a)$, i.e., $V(s,a(s)) = \min_{a \in \mathcal{A}} V(s,a)$. Symmetrically, the *best placement* with respect to a given attack $a \in \mathcal{A}$ (denoted by $s(a)$) is defined as any controllers placement s in $\mathcal{S}$ that maximizes the value of $V(s,a)$, i.e., $V(s(a),a) = \max_{s \in \mathcal{S}} V(s,a)$. In most cases, the so defined worst attacks and best placements are not unique.

The above described notation is summarized in Table 1.

3.2 | Problem description

Since the network operator is interested in maximizing the value of $V(s,a)$ while the attacker seeks to minimize it, both sides need to consider some kind of optimization approach for finding controllers placements (the operator) and for constructing attacks (the attacker). We assume that the set of possible controllers placements $\mathcal{S}$ and the set of possible attacks $\mathcal{A}$ are known to both the operator and the attacker, which leads to the following two problems.

Controller Placement Optimization Problem (CPOP)
Find a placement s^* whose resilience measure observed for its worst attack, i.e., $V(s^*, a(s^*))$, is the maximum over set $\mathcal{S}$:

$$V(s^*, a(s^*)) = \max_{s \in \mathcal{S}} V(s, a(s)) = \max_{s \in \mathcal{S}} \min_{a \in \mathcal{A}} V(s,a). \tag{5}$$

Each placement s^* that solves problem (5) is called the *best placement* for a given set of attacks $\mathcal{A}$. Clearly, such a placement s^* guarantees that the number of surviving nodes is at least $V(s^*, a(s^*))$ for any attack in $\mathcal{A}$, and $V(s^*, a(s^*))$ is the maximum number with this property.

Node Attack Optimization Problem (NAOP)
Find an attack a^* whose resilience measure observed for its best placement, i.e., $V(s(a^*), a^*)$, is the minimum over set $\mathcal{A}$:

$$V(s(a^*), a^*) = \min_{a \in \mathcal{A}} V(s(a), a) = \min_{a \in \mathcal{A}} \max_{s \in \mathcal{S}} V(s,a). \tag{6}$$

Each attack a^* that solves problem (6) is called the *worst attack* for a given set of placements $\mathcal{S}$. Thus, attack a^* guarantees that the number of surviving nodes is at most $V(s(a^*), a^*)$ for any placement in $\mathcal{S}$, and $V(s(a^*), a^*)$ is the minimum number with this property.

In this paper, which is a significant extension of the work presented in [33], we focus on the formulation and discussion of alternative optimization models dedicated to solving the Node Attack Optimization Problem, leaving the consideration of analogous models for the Controller Placement Optimization Problem for a future paper.

TABLE 1 Summary of notation.

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	network graph
$\mathcal{V}, \mathcal{E}$	sets of nodes and links, respectively ($V = \lvert\mathcal{V}\rvert, E = \lvert\mathcal{E}\rvert$)
$\alpha(e), \beta(e)$	end-nodes of link $e \in \mathcal{E}$
$\delta(v)$	set of links incident with node $v \in \mathcal{V}$
$\Delta(v)$	set of nodes adjacent to node $v \in \mathcal{V}$
$\deg(v)$	node degree ($\deg(v) = \lvert\delta(v)\rvert = \lvert\Delta(v)\rvert,\ v \in \mathcal{V}$)
$\mathcal{S}$	set of allowable controllers placements
$\mathcal{V}(s)$	set of controllers in placement s, ($V(s) = \lvert\mathcal{V}(s)\rvert,\ s \in \mathcal{S}$)
$\mathcal{A}$	set of considered attacks
$\mathcal{V}(a)$	set of nodes affected by attack $a \in \mathcal{A}$ ($V(a) = \lvert\mathcal{V}(a)\rvert,\ a \in \mathcal{A}$)
$\mathcal{G}(a)$	graph of the network affected by attack a
$\mathcal{C}(a)$	family of connected components induced by attack a ($C(a) = \lvert\mathcal{C}(a)\rvert,\ a \in \mathcal{A}$)
$\mathcal{C}(s, a)$	family of components in $\mathcal{C}(a)$ that contain at least one controller from placement s ($C(s, a) = \lvert\mathcal{C}(s, a)\rvert,\ s \in \mathcal{S}, a \in \mathcal{A}$)
$\overline{\mathcal{C}}(s, a)$	set (family) of components in $\mathcal{C}(a)$ that do not contain any controller from placement s ($\overline{C}(s, a) = \lvert\overline{\mathcal{C}}(s, a)\rvert,\ s \in \mathcal{S}, a \in \mathcal{A}$)
$\mathcal{V}(c)$	set of nodes of component $c \in \mathcal{C}(a)$ ($V(c) = \lvert\mathcal{V}(c)\rvert,\ c \in \mathcal{C}(a), a \in \mathcal{A}$)
$\mathcal{S}[M]$	set of all placements composed of M controllers
$\mathcal{A}[K]$	set of all K-node attacks
$V(s, a)$	number of nodes that survive attack $a \in \mathcal{A}$ when placement $s \in \mathcal{S}$ is assumed
$a(s)$	the worst attack in $\mathcal{A}$ for a given placement $s \in \mathcal{S}$
$s(a)$	the best placement in $\mathcal{S}$ for a given attack $a \in \mathcal{A}$
$\lvert\mathcal{X}\rvert$	number of elements in set $\mathcal{X}$ (size, cardinality)
$\mathcal{X}^{\lvert 2\rvert}$	set of all 2-element subsets of a given set $\mathcal{X}$
$\mathbb{B}$	set of binary numbers ($\mathbb{B} = \{0, 1\}$)
$\mathbb{Z}_+$	set of nonnegative integer numbers
$\mathbb{R}_+$	set of nonnegative real numbers

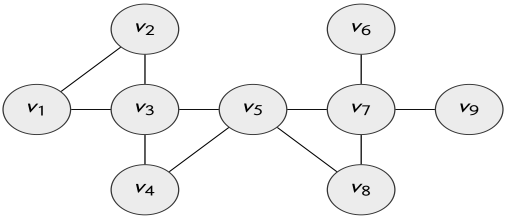

FIGURE 1 Sample 9-node and 11-link network.

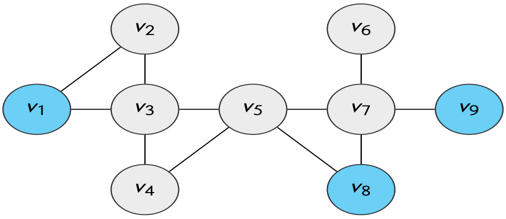

FIGURE 2 One of 32 best placements (with $\mathcal{V}(s) = \{1, 8, 9\}$).

3.3 | Example

Let us characterize the solutions of NAOP and CPOP for the network depicted in Figure 1 and the 3-controllers placements ($M = 3$) and the 2-node attacks ($K = 2$).

For this setting, in the set $\mathcal{A}[2]$ there are 2 worst attacks (out of all $\binom{9}{2} = 36$ attacks in $\mathcal{A}[2]$) with respect to $\mathcal{S}[3]$. Each of them guarantees that at most 6 nodes survive whatever 3-controllers placement is selected. These two attacks, a^1 and a^2, have the sets of attacked nodes equal to $\mathcal{V}(a^1) = \{3, 7\}$ and $\mathcal{V}(a^2) = \{5, 7\}$; the second of them is shown in Figure 3.

It turns out that in the set $\mathcal{S}[3]$ there are 32 best placements (out of all $\binom{9}{3} = 84$ placements in $\mathcal{S}[3]$) with respect to the set $\mathcal{A}[2]$ of all 2-node attacks. Each of them guarantees that at least 4 nodes survive whatever 2-node attack is selected. We do not list all those placements because there are too many of them; instead we show one of these placements (with the set of controller nodes $\{1, 8, 9\}$) in Figure 2.

Now let us assume that the attacker decides to use one of the two worst attacks to attack the network. Knowing that, the operator selects one of the best placements with respect to the set $\mathcal{A} = \{a^1, a^2\}$ and deploy it. Actually, there are four such best placements, s^1, s^2, s^3, s^4 with $\mathcal{V}(s^1) = \{1, 8, 9\}$, $\mathcal{V}(s^2) = \{2, 8, 9\}$, $\mathcal{V}(s^3) = \{1, 6, 8\}$, $\mathcal{V}(s^4) = \{2, 6, 8\}$. All of them belong to the set of 32 best placements with respect to the full set of the 2-node attacks, and

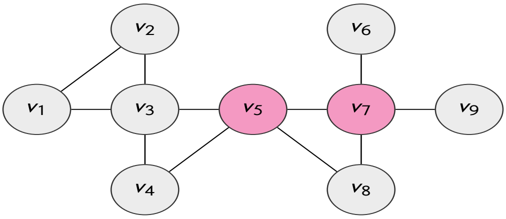

FIGURE 3 One of 2 worst attacks (with $\mathcal{V}(a) = \{5, 7\}$).

each of them guarantees that at least 6 nodes survive any of the two considered attacks. Notice that this value is clearly greater than 4, i.e., the number of surviving nodes guaranteed by the best placement with respect to the full set of the 2-node attacks. Moreover, there is no single 2-node attack ensuring that less than 6 nodes survive if any of the placements in the set $\mathcal{S} = \{s^1, s^2, s^3, s^4\}$ is deployed.

4 | NON-COMPACT FORMULATION OF NAOP

In the IP formulation of NAOP presented below (and introduced in [33]) we assume that $\mathcal{A} = \mathcal{A}[K]$ (i.e., we consider the set of all K-node attacks) and $\mathcal{S}$ is a given set of arbitrary controllers placements. This means that we consider the problem of finding a K-node attack that minimizes the number of surviving nodes when the best placement in the set $\mathcal{S}$ with respect to this attack is considered.

4.1 | NAOP formulation for a given set of controllers placements

In formulation $\mathbb{A}[K, \mathcal{S}]$ below, for each $v \in \mathcal{V}$, a_v is a binary variable equal to 1 if, and only if, node v is attacked. Each vector $a = (a_v)_{v\in\mathcal{V}}$ thus specifies an attack with $\mathcal{V}(a) = \{v \in \mathcal{V} : a_v = 1\}$. Next, for each $e \in \mathcal{E}$, t_e is a binary variable equal to 1 if, and only if, link e is not available as a result of the attack (i.e., one or both of its end-nodes are attacked). Finally, for each $s \in \mathcal{S}$ and $v \in \mathcal{V}$, z^s_v is a binary variable equal to 1 if, and only if, node v survives the constructed attack when controllers placement s is assumed. The formulation uses the fact that if a node can still provide service after the attack, then every node in its neighborhood (i.e., in a location interconnected with it by a transport link) can also provide service unless its location was directly attacked.

$$\mathbb{A}[K, \mathcal{S}] : \min Z \tag{7a}$$

$$\text{subject to } \sum_{v\in\mathcal{V}} a_v = K, \tag{7b}$$

$$t_e \geq a_v, \qquad v \in \mathcal{V},\ e \in \delta(v) \tag{7c}$$

$$t_e \leq a_{\alpha(e)} + a_{\beta(e)}, \qquad e \in \mathcal{E} \tag{7d}$$

$$z^s_v \leq 1 - a_v, \qquad s \in \mathcal{S},\ v \in \mathcal{V} \tag{7e}$$

$$z^s_v \geq 1 - a_v, \qquad s \in \mathcal{S},\ v \in \mathcal{V}(s) \tag{7f}$$

$$z^s_{\alpha(e)} \geq z^s_{\beta(e)} - t_e, \qquad s \in \mathcal{S},\ e \in \mathcal{E} \tag{7g}$$

$$z^s_{\beta(e)} \geq z^s_{\alpha(e)} - t_e, \qquad s \in \mathcal{S},\ e \in \mathcal{E} \tag{7h}$$

$$Z \geq \sum_{v\in\mathcal{V}} z^s_v, \qquad s \in \mathcal{S} \tag{7i}$$

$$a_v, t_e, z^s_v \in \mathbb{B}, \qquad s \in \mathcal{S},\ v \in \mathcal{V},\ e \in \mathcal{E} \tag{7j}$$

$$Z \in \mathbb{R}_+. \tag{7k}$$

Constraint (7b) sets the number of attacked nodes to K. Constraints (7c) and (7d) force the value of each binary variable t_e to be equal to 0, when link e is available after attack a, i.e., when both end-nodes of e are not directly attacked. Constraints (7e) set z^s_v to 0 when node v does not survive attack a, i.e., when node v is attacked directly, whatever placement is selected. Constraints (7f), in turn, set z^s_v to 1 if node v is not directly attacked and its location contains a controller.

The next two sets of constraints, (7g) and (7h), ensure that if link e is available after attack a (i.e., when $t_e = 0$), then its end-nodes either simultaneously survive or are simultaneously out of service (i.e., have no access to surviving controllers). This property ensures that in optimal solutions all nodes in any component $c \in \mathcal{C}(a)$ have the same values of z_v^s (for any fixed placement s):

$$z_v^s = z_w^s, \qquad v, w \in \mathcal{V}(c), c \in \mathcal{C}(a), s \in \mathcal{S}. \tag{8}$$

Moreover, for any given placement $s \in \mathcal{S}$, if a location $v \in \mathcal{V}(c)$ contains a controller then inequality (7f) sets z_v^s to 1 if this location is not attacked. In this case equalities (8) imply that the values of z_v^s are set to 1 for all $v \in \mathcal{V}(c)$, i.e., all nodes in c survive the attack, as required. In effect, for any $s \in \mathcal{S}$, all these nodes are counted in the summation on the right hand side of inequality (7i).

On the other hand, when component $c \in \mathcal{C}(a)$ does not contain any controller from placement s, then in the feasible solutions of the considered formulation all values of z_v^s, $v \in \mathcal{V}(c)$, are simultaneously equal either to 0 or to 1. This is not an issue, though. To see this consider an optimal solution of (7) and let $\mathcal{S}^*$ denote the set of all placements in $\mathcal{S}$ for which inequalities (7i) are binding, i.e., $Z^* = \sum_{v \in \mathcal{V}} z_v^s$ if, and only if, $s \in \mathcal{S}^*$, where Z^* is the optimal value of Z. Denoting the optimized attack by a^*, we can rewrite the equalities in question as follows:

$$Z^* = \sum_{v \in \mathcal{V}} z_v^s = \sum_{c \in \mathcal{C}(a^*)} \sum_{v \in \mathcal{V}(c)} z_v^s, \qquad s \in \mathcal{S}^*, \tag{9}$$

because $\mathcal{V} = \mathcal{V}(a^*) \cup \bigcup_{c \in \mathcal{C}(a^*)} \mathcal{V}(c)$ and $z_v^s = 0$ for $v \in \mathcal{V}(a^*)$, i.e., when $a_v = 1$. This shows that for each $s \in \mathcal{S}^*$, the sum $\sum_{v \in \mathcal{V}(c)} z_v^s$ must be equal to 0 for all components $c \in \mathcal{C}(a^*)$ that do not contain a controller from placement s. Otherwise, Z^* would not be minimal since if any of such sums were greater than 0, then setting it to 0, which is allowed by the constraints, would result in a feasible solution with $Z < Z^*$. Note that this reasoning reveals that constraints (7e) are redundant.

In summary, the minimum value Z^* of objective (7a) is equal to

$$V(s(a^*), a^*) = \min_{a \in \mathcal{A}[K]} \max_{s \in \mathcal{S}} V(s, a), \tag{10}$$

where a^* denotes an optimal attack resulting from (7), that is one of the worst attacks in $\mathcal{A}[K]$ for the assumed set of placements $\mathcal{S}$.

4.2 | Solving $\mathbb{A}[K, \mathcal{S}[M]]$ by controllers placements generation

Formulation $\mathbb{A}[K, \mathcal{S}]$ is in general non-compact as the number of placements in the set $\mathcal{S}$ may grow exponentially with the number of nodes V. When this is the case, NAOP can be approached using a controllers placement generation procedure (provided $\mathcal{S}$ can be characterized in a tractable way).

Such non-compactness can appear for $\mathcal{S} = \mathcal{S}[M]$, for example when $V = 2M$. Therefore, below we present an iterative algorithm dedicated to solving the problem of finding the worst attack for the full list of M-node controllers placements $\mathcal{S} = \mathcal{S}[M]$, i.e., to solving formulation $\mathbb{A}[K, \mathcal{S}[M])]$. This algorithm, referred to as CPGA (Controller Placement Generation Algorithm), is a modification of algorithm A2 presented in [33].

In CPGA, the list of placements $\mathcal{S}$ is iteratively extended by means of finding the best placement $s(a^*)$ for consecutive attacks a^* obtained by solving $\mathbb{A}[K, \mathcal{S}]$ for the current list $\mathcal{S}$. In effect, in each iteration we find out whether there exists a placement $s \in \mathcal{S}[M] \setminus \mathcal{S}$ such that when s is added to the current list of placements $\mathcal{S}$, then the number of surviving nodes for s is greater than the minimum number of surviving nodes guaranteed for any placement in $\mathcal{S}$ (achieved for the current optimal attack). If this is the case, we re-optimize a^* and continue.

The following algorithm solves NAOP for the set of attacks $\mathcal{A}[K]$ and the set of controllers placements $\mathcal{S}[M]$, i.e., problem $\mathbb{A}[K, \mathcal{S}[M]]$.

CPGA: Algorithm for attack optimization by controllers placements generation

Step 0**:** Generate a random K-node attack a^*; $\mathcal{S} := \emptyset$, $Z^* := 0$
(Comment: for $\mathcal{S} = \emptyset$ any attack in $\mathcal{A}[K]$ solves $\mathbb{A}[K, \mathcal{S}]$ giving $Z^* = 0$.)
Step 1**:** Find the best placement $s^* = s(a^*)$ (assuring Y^* surviving nodes) with respect to attack a^*. If $Y^* \leq Z^*$ then go to Step 3.
Step 2**:** $\mathcal{S} := \mathcal{S} \cup \{s^*\}$. Solve $\mathbb{A}[K, \mathcal{S}]$ to get the worst attack a^* (assuring at most Z^* surviving nodes) with respect to set $\mathcal{S}$. Go to Step 1.
(Comment: Z^* is equal to $V(s(a^*), a^*) = \min_{a \in \mathcal{A}[K]} \max_{s \in \mathcal{S}} V(s, a)$.)
Step 3**:** Stop: current attack a^* is an optimal solution of $\mathbb{A}[K, \mathcal{S}[M]]$ that is

$$Z^* = V(s(a^*), a^*) = \min_{a \in \mathcal{A}[K]} \max_{s \in \mathcal{S}[M]} V(s, a).$$

Note that finding the best placement $s^* = s(a^*)$ in Step 1 is straightforward. When $M > C(a^*)$, i.e., when the number of controllers is greater than the number of components in $\mathcal{C}(a^*)$, then $Y^* = V - V(a^*)$ and the controllers can be located anywhere in the set of surviving nodes, provided that each component contains at least one controller. Otherwise, Y^* is equal to the sum of the sizes of the M largest components in $\mathcal{C}(a^*)$, and s^* is obtained by locating one controller in each of them. Note that the components in $\mathcal{C}(a^*)$ can be found very quickly by performing depth-first search in the network graph induced by the set of nodes $\mathcal{V} \setminus \mathcal{V}(a)$. This observation will be further exploited in Section 5.1.

5 | COMPACT FORMULATIONS OF NAOP

The compact IP formulations described in this section are designed for directly solving the NAOP problem, and are alternatives to the iterative algorithm CPGA presented in Section 4.2. Recall that CPGA uses a non-compact formulation (7) of NAOP, i.e., a formulation that assumes a predefined list $\mathcal{S}$ of controllers placements for which the actual sets of controllers locations $\mathcal{V}(s)$, $s \in \mathcal{S}$, are known (they are used in constraints (7f)). The compact formulations are of interest as CPGA does not scale well with the size of the network.

The NAOP formulations considered in this section assume that $\mathcal{A} = \mathcal{A}[K]$ and $\mathcal{S} = \mathcal{S}[M]$ for a given pair K, M of non-negative integers. Note that the same assumption was made in CPGA.

5.1 | Pseudo-component approach

Below we present three compact IP formulations, $\mathbb{F}1[K, M]$, $\mathbb{F}2[K, M]$, $\mathbb{F}3[K, M]$, of the min-max node attack optimization problem (6) that, unlike their non-compact counterpart $\mathbb{A}[K, \mathcal{S}[M]]$ specified by formulation (7), do not require an explicit list $\mathcal{S}[M]$ of the M-node controllers placements.

The formulations are based on the observation made at the end of Section 4.2. If $M \leq C(a)$ (i.e., when the number of controllers is smaller than or equal to the number of components) then the best M-node controllers placement $s(a)$ is obtained by allocating one controller to each of the M largest components in $\mathcal{C}(a)$, and if $C(a) < M$ (i.e., when the number of components is smaller than the number of controllers) then the best placement allocates one controller in each component, while the remaining $M - C(a)$ controllers are allocated anywhere (or not allocated at all); in this way all nodes that are not directly attacked, i.e., the nodes in the set $\mathcal{V} \setminus \mathcal{V}(a)$, survive.

5.1.1 | Pseudo-components and related notions

Let us consider a given attack $a \in \mathcal{A}$ and the family $\mathcal{C}(a)$ of the connected components induced by this attack. A set of nodes $\mathcal{P}$ is called a *pseudo-component* (*p-component* in short) induced by attack a if, and only if, $\mathcal{P} = \bigcup_{c \in \mathcal{C}'} \mathcal{V}(c)$ for some subfamily of components $\mathcal{C}' \subseteq \mathcal{C}(a)$. Thus, a p-component $\mathcal{P}$ can be either empty ($\mathcal{C}' = \emptyset$), equal to $\mathcal{V}(c)$ for some $c \in \mathcal{C}(a)$ ($|\mathcal{C}'| = 1$) and then called a *proper pseudo-component*, or composed of more than one component ($2 \leq |\mathcal{C}'| \leq C(a)$) and then called a *non-proper pseudo-component*. Clearly, non-proper p-components are not connected in the graph affected by attack a since the components in $\mathcal{C}(a)$ are maximal connected subgraphs of graph $\mathcal{G}(a)$.

Now let us consider an indexed family $\mathcal{X} = \{\mathcal{X}^n, n \in \mathcal{N}\}$ of p-components induced by attack a, where $\mathcal{N} = \{1, 2, \ldots, N\}$, and $C(a) \leq N \leq V - V(a)$ (recall that $V(a)$ is the number of nodes attacked by a). Then $\mathcal{X}$ is called a N-family induced by attack a if, and only if, the p-components $\mathcal{X}^n$ are pairwise disjoint and for each component $c \in \mathcal{C}(a)$, $\mathcal{V}(c) \subseteq \mathcal{X}^n$ for some $n \in \mathcal{N}$. The set of all such defined N-families induced by attack a will be denoted by $\mathcal{F}(a, N)$. In the following, N-families containing only proper and empty p-components will be called *proper*, and *non-proper* otherwise. The vector of the sizes of the p-components in $\mathcal{X}$ will be denoted by $X = (X^n)_{n \in \mathcal{N}}$, where $X^n = |\mathcal{X}^n|$, $n \in \mathcal{N}$. Note that there can be more than one N-families with the same vector X.

If $N \geq C(a)$ then each proper N-family induced by attack a contains all proper p-components $\mathcal{V}(c)$, $c \in \mathcal{C}(a)$, plus $N - C(a)$ empty p-components (this means that in total there are $\frac{N!}{(N-C(a))!}$ proper N-families in $\mathcal{F}(a, N)$). An example of a non-proper N-family is a family composed of just one p-component $\mathcal{P} = \bigcup_{c \in \mathcal{C}(a)} \mathcal{V}(c) = \mathcal{V} \setminus \mathcal{V}(a)$ and N–1 empty p-components. Clearly, in general $\mathcal{F}(a, N)$ contains many other N-families between these two extremities.

Note that for any attack $a \in \mathcal{A}$, the value of the measure $V(s(a), a)$ (see (4)) for its best controllers placement $s(a)$ composed of M nodes is equal to the total number of nodes in the M largest components in $\mathcal{C}(a)$, i.e., to the total number of nodes in the M largest p-components in any proper N-family in $\mathcal{F}(a, N)$. At the same time, $V(s(a), a)$ is the lower bound on the total number of nodes in the M largest p-components in any non-proper N-family in $\mathcal{F}(a, N)$.

To see this, consider a non-proper family $\mathcal{X}$ in $\mathcal{F}(a, N)$ and denote by $\Theta(\mathcal{X})$ the total number of nodes in its M largest p-components. Let $\mathcal{Y}$ be a subfamily of $\mathcal{X}$ composed of the M largest p-components in $\mathcal{X}$ (note that such subfamily is in general not unique, and, by definition, the total number of nodes in the p-components in $\mathcal{Y}$ is equal to $\Theta(\mathcal{X})$). Further, consider an arbitrary non-proper p-component $\mathcal{X}^i \in \mathcal{X}$ and one of the empty p-components $\mathcal{X}^j$ (note that every non-proper family in $\mathcal{F}(a, N)$ contains at least one empty p-component since the number of its non-empty p-components is strictly less than $C(a)$ and $C(a) \leq N$). Now we split $\mathcal{X}^i$ into two non-empty p-components, $\mathcal{P}'$ and $\mathcal{P}''$, say, and construct a new N-family $\mathcal{X}' \in \mathcal{F}(a, N)$ that consists of the same elements as $\mathcal{X}$ for $n \in \mathcal{N} \setminus \{i, j\}$, p-component $\mathcal{P}'$ instead of $\mathcal{X}^i$ and p-component $\mathcal{P}''$ instead of $\mathcal{X}^j$. Finally, we consider two cases: $\mathcal{X}^i \in \mathcal{X} \setminus \mathcal{Y}$ and $\mathcal{X}^i \in \mathcal{Y}$. In the first case $\Theta(\mathcal{X}') = \Theta(\mathcal{X})$ since $\mathcal{Y}$ is a subfamily of $\mathcal{X}'$ composed of its M largest p-components as well. In the second case $\Theta(\mathcal{X}') \leq \Theta(\mathcal{X})$ because of the following reason. Let $\mathcal{X}^k$ be the largest p-component in $\mathcal{X} \setminus \mathcal{Y}$. Since the size of $\mathcal{X}^k$ is not greater than the size of any p-component in $\mathcal{Y}$, a subfamily $\mathcal{Y}'$ of $\mathcal{X}'$ composed of the M largest p-components in $\mathcal{X}'$ is obtained by replacing $\mathcal{X}^i$ in $\mathcal{Y}$ with the largest p-component among $\mathcal{P}', \mathcal{P}'', \mathcal{X}^k$ (note that $\Theta(\mathcal{X}')$ is strictly less than $\Theta(\mathcal{X})$ if the size of $\mathcal{X}^i$ is strictly greater than the size of $\mathcal{X}^k$).

The above construction applied iteratively will end up with a proper N-family, which shows that $\Theta(\mathcal{X}) \geq V(s(a), s)$ for any $\mathcal{X} \in \mathcal{F}(a, N)$. However, as illustrated below, this does not preclude the possibility of achieving $\Theta(\mathcal{X}) = V(s(a), a)$ by some non-proper N-family $\mathcal{X}$ in $\mathcal{F}(a, N)$.

Example: Consider an attack a for which family $C(a)$ is composed of 3 components of sizes $10, 6, 4$, and a proper N-family $\hat{\mathcal{X}} = \{\hat{\mathcal{X}}^1, \hat{\mathcal{X}}^2, \hat{\mathcal{X}}^3, \hat{\mathcal{X}}^4\}$ in $\mathcal{F}(a, N)$ (where $N = 4$) with $\hat{X} = (\hat{X}^1, \hat{X}^2, \hat{X}^3, \hat{X}^4) = (10, 6, 4, 0)$. This means that the first three elements in $\hat{\mathcal{X}}$ are proper p-components and the last one is an empty p-component. The following are examples of non-proper N-families in $\mathcal{F}(a, N)$ for which the sum of their M largest p-components is equal to that of the proper N-family $\hat{\mathcal{X}}$:

- $M = 1$. In this case $V(s(a), a)$ is equal to 10, i.e., to the size of the largest component in $C(a)$. The same value is obtained for the non-proper 4-family defined as follows:
  - $\mathcal{X}^1 = \hat{\mathcal{X}}^1, \mathcal{X}^2 = \hat{\mathcal{X}}^2 \cup \hat{\mathcal{X}}^3, \mathcal{X}^3 = \mathcal{X}^4 = \varnothing,$ $X^1 = 10, X^2 = 10, X^3 = X^4 = 0.$

    The first p-component is proper, the second is composed of the two remaining components in $C(a)$, and the next two p-components are empty.

- $M = 3$. Now $V(s(a), a)$ is equal to 20, i.e., to the sum of the sizes of all three components in $C(a)$. The same value is obtained for each of the following three non-proper 4-families:
  - $\mathcal{X}^1 = \hat{\mathcal{X}}^1 \cup \hat{\mathcal{X}}^2 \cup \hat{\mathcal{X}}^3, \mathcal{X}^2 = \mathcal{X}^3 = \mathcal{X}^4 = \varnothing,$ $X^1 = 20, X^2 = X^3 = X^4 = 0$
  - $\mathcal{X}^1 = \hat{\mathcal{X}}^1 \cup \hat{\mathcal{X}}^2, \mathcal{X}^2 = \hat{\mathcal{X}}^3, \mathcal{X}^3 = \mathcal{X}^4 = \varnothing,$ $X^1 = 16, X^2 = 4, X^3 = X^4 = 0$
  - $\mathcal{X}^1 = \hat{\mathcal{X}}^1 \cup \hat{\mathcal{X}}^3, \mathcal{X}^2 = \hat{\mathcal{X}}^2, \mathcal{X}^3 = \mathcal{X}^4 = \varnothing,$ $X^1 = 14, X^2 = 6, X^3 = X^4 = 0.$

As will soon become apparent, the above property is essential for the correctness (and efficiency) of the formulations presented in the next three sections.

Let $N[K] = \max_{a \in \mathcal{A}[K]} C(a)$ denote the maximum number of connected components into which the network graph can be split by the attacks in $\mathcal{A}[K]$. Note that $N[K]$ is the smallest number N that ensures that for each attack $a \in \mathcal{A}[K]$ there will be at least one proper family in $\mathcal{F}(a, N)$. The value of $N[K]$ can be computed by means of formulation (16) described in Section 5.1.4 and used for parameter N in the problem formulation presented below.

5.1.2 | Formulation $\mathbb{F}1\,[K, M]$

The first compact IP formulation of NAOP making use of the notion of pseudo-component assumes that $N[K] \leq N \leq V - K$, and is as follows.

Formulation $\mathbb{F}1\,[K, M]$:

$$\begin{aligned}
\text{minimize} \quad & Z = \sum_{n=1}^{M} X^n && && (11a)\\
\text{subject to} \quad & \sum_{v \in \mathcal{V}} a_v = K && && (11b)\\
& a_v + \sum_{n \in \mathcal{N}} x_v^n = 1, && v \in \mathcal{V} && (11c)\\
& X^n = \sum_{v \in \mathcal{V}} x_v^n, && n \in \mathcal{N} && (11d)\\
& x_{\alpha(e)}^n + \sum_{m \in \mathcal{N} \setminus \{n\}} x_{\beta(e)}^m \leq 1, && e \in \mathcal{E},\ n \in \mathcal{N} && (11e)\\
& X^1 \geq X^2 \geq \cdots \geq X^N && && (11f)\\
& a_v, x_v^n \in \mathbb{B},\ X^n \in \mathbb{Z}_+, && v \in \mathcal{V},\ n \in \mathcal{N}. && (11g)
\end{aligned}$$

Above, similarly to formulation (7), the vector of binary variables $a = (a_v)_{v \in \mathcal{V}}$ specifies the attack, i.e., it is the characteristic vector of the set of nodes $\mathcal{V}(a) = \{v \in \mathcal{V} : a_v = 1\}$ that are attacked, and constraint (11b) sets the number of attacked nodes to K.

Let us assume for a while that a is fixed. Then binary variables x_v^n, $v \in \mathcal{V}$, $n \in \mathcal{N}$ determine an indexed family of sets $\mathcal{X} = \{\mathcal{X}^n, n \in \mathcal{N}\}$, where $\mathcal{X}^n = \{v \in \mathcal{V} : x_v^n = 1\}$ which, due to constraints (11c), is composed of N pairwise disjoint (but not necessarily non-empty) sets that form a partition of the set $\mathcal{V} \setminus \mathcal{V}(a)$ of nodes that are not directly affected by attack a. Additionally, constraints (11d) determine, for each set $\mathcal{X}^n$, the number of its elements $X^n = |\mathcal{X}^n|$. Next, constraints (11e) (called common p-component constraints in the following) forbid the nodes belonging to different p-components in family $\mathcal{X}$ to be connected by any link that survives the considered attack. This ensures that non-empty sets in $\mathcal{X}$ are p-components, in general non-proper since such sets are not necessarily connected (there are no constraints forcing that).

Thus, each so defined family $\mathcal{X}$ is an N-family for attack a, i.e., $\mathcal{X} \in \mathcal{F}(a, N)$, since each set $\mathcal{X}^n \in \mathcal{X}$ is either empty or non-empty and not connected (then it is a non-proper p-component), or non-empty and connected (then it is a proper p-component). Moreover, all possible N-families in $\mathcal{F}(a, N)$ are feasible with respect to constraints (11c)-(11e).

The last (monotonicity) constraint (11f) arranges the p-components in family $\mathcal{X}$ according to their non-increasing sizes, and due to that the objective function in (11a) expresses the total number of nodes in the M largest p-components in the constructed N-family $\mathcal{X}$.

It follows that if a is fixed, optimizing the objective function over variables x_v^n, $v \in \mathcal{V}, n \in \mathcal{N}$, results in finding a N-family $\mathcal{X}(a)$ which minimizes $\Theta(\mathcal{X})$ over $\mathcal{X} \in \mathcal{F}(a, N)$, so that $\Theta(\mathcal{X}(a)) = V(s(a), a)$. In effect, when we finally let a_v, $v \in \mathcal{V}$, become variables then we find an attack $a^* \in \mathcal{A}[K]$ that minimizes the value of measure $V(s(a), a)$ (see (6)), i.e., an attack that solves the considered version of NAOP (for K-node attacks and M-node placements).

As explained in the previous section, in general the N-family $\mathcal{X}(a^*)$ determined by the optimal values of variables x_v^n, $v \in \mathcal{V}, n \in \mathcal{N}$, is not necessarily proper. This, however, is not an issue since family $C(a^*)$ can be easily determined by performing depth-first search on each p-component to split it, if needed, into proper p-components.

5.1.3 | Formulation $\mathbb{F}2[K, M]$

The second compact formulation of NAOP, $\mathbb{F}2[K, M]$, presented below differs from the first one, $\mathbb{F}1[K, M]$, in only one respect. This time the monotonicity constraint (11f) is not used and the sum of the M largest p-components (i.e., objective function) is found in a different way. The formulation is as follows.

Formulation $\mathbb{F}2[K, M]$:

$$
\begin{aligned}
\text{minimize} \quad & Z = MB + \sum_{n \in \mathcal{N}} b^n && && \text{(12a)}\\
\text{subject to} \quad & \sum_{v \in \mathcal{V}} a_v = K && && \text{(12b)}\\
& a_v + \sum_{n \in \mathcal{N}} x_v^n = 1, && v \in \mathcal{V} && \text{(12c)}\\
& X^n = \sum_{v \in \mathcal{V}} x_v^n, && n \in \mathcal{N} && \text{(12d)}\\
& x_{\alpha(e)}^n + \sum_{m \in \mathcal{N} \setminus \{n\}} x_{\beta(e)}^m \le 1, && e \in \mathcal{E},\, n \in \mathcal{N} && \text{(12e)}\\
& b^n \ge X^n - B, && n \in \mathcal{N} && \text{(12f)}\\
& a_v, x_v^n \in \mathbb{B},\ b^n, X^n, B \in \mathbb{Z}_+, && v \in \mathcal{V},\, n \in \mathcal{N}. && \text{(12g)}
\end{aligned}
$$

In the above formulation, all variables and constraints that are common with formulation (11) have the same meaning. What is new, are variables B and b^n, $v \in \mathcal{V}$, constraints (12f) and the objective function in (12a).

In fact, for a given fixed sequence $X^1, X^2, \ldots, X^N$ of N numbers, the minimum of $MB + \sum_{n \in \mathcal{N}} b^n$ constrained by (12f) and taken over non-negative continuous variables B and b^n, $n \in \mathcal{N}$, is equal to the sum of the M largest numbers in the considered sequence. This fact can be proved directly, but the easiest way to derive it is to consider the following linear program.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{n \in \mathcal{N}} X^n u^n && && \text{(13a)}\\
\text{subject to} \quad & && && \\
[B \ge 0] \quad & \sum_{n \in \mathcal{N}} u^n \le M && && \text{(13b)}\\
[b^n \ge 0] \quad & u^n \le 1, && n \in \mathcal{N} && \text{(13c)}\\
& u^n \in \mathbb{R}_+, && n \in \mathcal{N}. && \text{(13d)}
\end{aligned}
$$

Clearly, the optimal solutions of this formulation determine the sum of the M largest numbers in the sequence in question. It is also obvious that vertex solutions of (13) are binary. Now, we formulate the dual of (13) using the dual variables listed above in square brackets on the left hand side of the corresponding constraints. The dual formulation is as follows.

$$\begin{aligned} \text{minimize} \quad & MB + \sum_{n \in \mathcal{N}} b^n && && \text{(14a)} \\ \text{subject to} \quad & b^n \geq X^n - B, && n \in \mathcal{N} && \text{(14b)} \\ & b^n, B \in \mathbb{R}_+, && n \in \mathcal{N}. && \text{(14c)} \end{aligned}$$

Note that when $X^1, X^2, \ldots, X^N$ are integer, then the vertex solutions of the dual are also integer. It follows that formulation $\mathbb{F}2[K, M]$ is correct. (In fact, assuming that dual variables b^n, B in $\mathbb{F}2[K, M]$ are continuous is also correct.)

Finally, note that the discussion at the end of Section 5.1.2 also applies to formulation $\mathbb{F}2[K, M]$.

5.1.4 | Formulation $\mathbb{F}3[K, M]$

An issue in formulation $\mathbb{F}2[K, M]$ is the significant amount of symmetry, as all permutations of any feasible N-family $\mathcal{X} = \{\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^N\}$ are also feasible. To alleviate this, we introduce the third compact NAOP formulation that uses the notion of a *representative* of the class of symmetrical solutions. This approach was introduced in [4] for the vertex coloring problem and used e.g. in [5] in the context of a network design application.

In formulation $\mathbb{F}3[K, M]$ we take advantage of the fact that the nodes are numbered (recall that $\mathcal{V} = \{1, 2, \ldots, V\}$) and thus ordered, and assume that $N = V$. Thus, since $\mathcal{N} = \{1, 2, \ldots, V\}$, the sets $\mathcal{V}$ and $\mathcal{N}$ of the indices of nodes and p-components are equal. The major difference with the previous formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$ is that we force each non-empty p-component $\mathcal{X}^n$ to contain node n and no node with index less than n. This property breaks the symmetry since each element (p-component) $\mathcal{X}^n$ of a feasible N-family is "represented" by the node with the smallest index, i.e., $n = \min\{m \in \mathcal{V} : x_m^n = 1\}$. In fact, this is the reason why the number of p-components N in the considered N-families must be equal to V, as it can happen than the last node (i.e., node indexed by V) constitutes the p-component of the form $\{v_V\}$.

The formulation is as follows.

Formulation $\mathbb{F}3[K, M]$:

$$\begin{aligned} \text{minimize} \quad & Z = MB + \sum_{n \in \mathcal{N}} b^n && && \text{(15a)} \\ \text{subject to} \quad & \sum_{v \in \mathcal{V}} a_v = K && && \text{(15b)} \\ & a_v + \sum_{n=1}^{v} x_v^n = 1, && v \in \mathcal{V} && \text{(15c)} \\ & x_n^n \geq x_v^n, && n \in \mathcal{N},\ v \in \mathcal{V},\ v \geq n && \text{(15d)} \\ & X^n = \sum_{v=n}^{N} x_v^n, && n \in \mathcal{N} && \text{(15e)} \\ & x_{\alpha(e)}^n + \sum_{m=1}^{n-1} x_{\beta(e)}^m + \sum_{m=n+1}^{\beta(e)} x_{\beta(e)}^m \leq 1, && e \in \mathcal{E},\ n \in \mathcal{N},\ n \leq \alpha(e) && \text{(15f)} \\ & b^n \geq X^n - B, && n \in \mathcal{N} && \text{(15g)} \end{aligned}$$

$$a_v \in \mathbb{B}, \qquad v \in \mathcal{V} \tag{15h}$$

$$b^n, X^n, B \in \mathbb{Z}_+, \qquad n \in \mathcal{N} \tag{15i}$$

$$x_v^n \in \mathbb{B}, \qquad v \in \mathcal{V},\ n = 1, 2, \ldots, v. \tag{15j}$$

The symmetry breaking property is ensured by two sets of constraints: constraints (15c) (where the range of summation differs from the range used in analogous constraints in $\mathbb{F}[K, M]$ and $\mathbb{F}2[K, M]$) and constraints (15d). The first set implies that each node indexed by v that is not directly attacked must belong to one of the p-components with the index not greater than v. Note that this implies $x_v^n = 0, n > v$, and hence these variables are not used in the formulation, see (15j); in effect, the number of variables x_v^n used in the formulation is equal to $\frac{V(V+1)}{2}$. The second set of constraints (15d), in turn, implies that when $x_n^n = 0$ (i.e., $n \notin \mathcal{X}^n$), then the p-component $\mathcal{X}^n$ is empty. Altogether, this means that if $\mathcal{X}^n$ is non-empty, then it must contain the node with index n and it cannot contain nodes with smaller indices. This is taken into account in constraints (15f) to reduce the range of n and m as compared with analogous constraints in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$. Note that the important assumption here is $\alpha(e) < \beta(e),\ e \in \mathcal{E}$.

Formulation $\mathbb{F}3[K, M]$ can be modified to pre-compute the values of $N[K] = \max_{a \in \mathcal{A}[K]} C(a)$ (i.e., the maximum number of connected components induced by K-node attacks assumed in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$) in the following way.

$$\text{maximize} \quad \sum_{n \in \mathcal{N}} y^n \tag{16a}$$

$$\text{subject to} \quad \sum_{v \in \mathcal{V}} a_v = K \tag{16b}$$

$$a_v + \sum_{n=1}^{v} x_v^n = 1, \qquad v \in \mathcal{V} \tag{16c}$$

$$x_n^n \geq x_v^n, \qquad n \in \mathcal{N},\ v \in \mathcal{V},\ v \geq n \tag{16d}$$

$$X^n = \sum_{v=n}^{N} x_v^n, \qquad n \in \mathcal{N} \tag{16e}$$

$$y^n \leq X^n, \qquad n \in \mathcal{N} \tag{16f}$$

$$x_{\alpha(e)}^n + \sum_{m=1}^{n-1} x_{\beta(e)}^m + \sum_{m=n+1}^{\beta(e)} x_{\beta(e)}^m \leq 1, \qquad e \in \mathcal{E},\ n \in \mathcal{N},\ n \leq \alpha(e) \tag{16g}$$

$$a_v \in \mathbb{B}, \qquad v \in \mathcal{V} \tag{16h}$$

$$X^n \in \mathbb{Z}_+,\ y^n \in \mathbb{B}, \qquad n \in \mathcal{N} \tag{16i}$$

$$x_v^n \in \mathbb{B}, \qquad v \in \mathcal{V},\ n = 1, 2, \ldots, v. \tag{16j}$$

Note that here parameter M and variables B, b^n, $n \in \mathcal{N}$, are not used, and additional (binary) variables y^n, $n \in \mathcal{N}$, indicate whether or not a given p-component n is non-empty. More precisely, constraints (16f) together with objective (16a) force y^n to be equal to 1 when p-component n is non-empty, and to 0, otherwise. Clearly, the maximum of the objective function is equal to $N[K]$.

5.2 | Bilevel approach

Below we describe an IP formulation $\mathbb{F}4[K, M]$, which is an alternative to the three compact IP formulations presented above. $\mathbb{F}4[K, M]$ is derived from a bilevel model for NAOP (for surveys on bilevel programming see for example [6, 9, 16]). The leader in this approach is the attacker who selects an attack in order to minimize the number of surviving nodes, while the follower (i.e., the network operator) selects the controller locations, knowing the attack, in order to maximize the number of surviving nodes. This leads to the following model.

$$
\begin{aligned}
&\text{minimize} \quad && Z = \Phi(a) && && \text{(17a)}\\
&\text{subject to} \quad && \sum_{v \in \mathcal{V}} a_v \leq K && && \text{(17b)}\\
& && a_v \in \mathbb{B}, && v \in \mathcal{V} && \text{(17c)}\\
&\text{where } \Phi(a) = \max && \sum_{v \in \mathcal{V}} \left(s_v + \sum_{j \in \Delta(v)} z^v_{vj} \right) && && \text{(17d)}\\
&\text{subject to} \quad && s && && \\
&[B \geq 0] && \sum_{v \in \mathcal{V}} s_v \leq M && && \text{(17e)}\\
&[y^v_i \geq 0] && \sum_{j \in \Delta(i)} \left(z^v_{ij} - z^v_{ji} \right) \geq -s_i, && i, v \in \mathcal{V},\ i \neq v && \text{(17f)}\\
&[t^v_i \geq 0] && \sum_{j \in \Delta(i)} z^v_{ij} + s_i \leq 1 - a_i, && i, v \in \mathcal{V} && \text{(17g)}\\
& && s_v \in \mathbb{B}, && v \in \mathcal{V} && \text{(17h)}\\
& && z^v_{iv} = 0, && \{i, v\} \in \mathcal{E} && \text{(17i)}\\
& && z^v_{ij}, z^v_{ji} \in \mathbb{R}_+, && \{i, j\} \in \mathcal{E},\ v \in \mathcal{V}. && \text{(17j)}
\end{aligned}
$$

The leader's problem (17a)-(17c) consists of choosing up to K nodes to attack (constraint (17b)), identified by binary variables $(a_v)_{v \in \mathcal{V}}$. The leader's objective (17a) is to minimize, over all attacks of at most K nodes, the value of function $\Phi(a)$ that counts the number of surviving nodes after the follower selects an optimal controllers placement for a given attack a.

To model the follower's problem (17d)-(17j) for a fixed attack a, we use binary variables $(s_v)_{v \in \mathcal{V}}$ characterizing the constructed controllers placement with up to M nodes (constraint (17e)). To check if a given node $v \in \mathcal{V}$ is connected to a surviving controller, we introduce non-negative continuous link flow variables z^v_{ij}, such that only the flows outgoing from v can be positive (constraints (17i)).

Then, constraints (17g) imply that a controller cannot be placed at an attacked node and that no flow leaves a controller or an attacked node. For all other nodes v, they impose the upper bound equal to 1 on the total flow outgoing from v. If a flow leaves node v, constraints (17f) make sure that this flow ends at locations equipped with a controller. Hence, the presence of a flow of value 1 going out of v indicates that there exists a path from v to a controller in the subgraph surviving the attack, and therefore that node v survives. The objective of (17d) is to maximize the total flow sent from the nodes, which is equal to the number of surviving nodes that are not controllers, plus the number of controllers. Since we forbid to install controllers at the attacked nodes (constraints (17g)), a controller always survives and hence $\Phi(a)$ is equal to the total number of surviving nodes.

Theorem 1 *Given a fixed attack $a = (a_v)_{v\in\mathcal{V}}$, the linear relaxation of the MIP formulation* (17d)-(17j) *of the follower's problem admits an optimal solution $s = (s_v)_{v\in\mathcal{V}}$ where $s_v \in \mathbb{B}$ for all $v \in \mathcal{V}$.*

The proof of Theorem 1 is given in Appendix A.

Corollary 2 *Theorem 1 implies that in order to compute the maximum of $\Phi(a)$ for a given attack a, we can replace the linear relaxation of the follower's problem formulation* (17d)-(17j) *by its dual, as the minimum of the dual is equal to the maximum of $\Phi(a)$.*

Thus, let us consider the dual of the linear relaxation of formulation (17d)-(17j) for a fixed vector a. The formulation of the dual uses the dual variables introduced on the left hand sides of constraints (17e)-(17g) and is as follows.

$$\begin{aligned}
&\text{minimize} && MB + \sum_{i\in\mathcal{V}}\sum_{v\in\mathcal{V}}(1-a_i)t_i^v && && \text{(18a)}\\
&\text{subject to} && && && \\
&[s_i \geq 0] && B - \sum_{v\in\mathcal{V}\setminus\{i\}} y_i^v + \sum_{v\in V} t_i^v \geq 1, && i \in \mathcal{V} && \text{(18b)}\\
&[z_{vj}^v \geq 0] && t_v^v + y_j^v \geq 1, && \{v,j\} \in \mathcal{E} && \text{(18c)}\\
&[z_{ij}^v \geq 0] && t_i^v + y_j^v - y_i^v \geq 0, && \{i,j\} \in \mathcal{E}, v \in \mathcal{V}\setminus\{i,j\} && \text{(18d)}\\
&[z_{ji}^v \geq 0] && t_j^v + y_i^v - y_j^v \geq 0, && \{i,j\} \in \mathcal{E}, v \in \mathcal{V}\setminus\{i,j\} && \text{(18e)}\\
& && y_i^v \in \mathbb{R}_+, && i, v \in \mathcal{V}, i \neq v && \text{(18f)}\\
& && t_i^v \in \mathbb{R}_+, && i, v \in \mathcal{V}; && \text{(18g)}\\
& && B \in \mathbb{R}_+. && && \text{(18h)}
\end{aligned}$$

Theorem 3 *Given a fixed attack $a = (a_v)_{v\in\mathcal{V}}$, the dual formulation* (18) *admits an optimal solution B, $t = (t_i^v)_{v,i\in\mathcal{V}}$ and $y = (y_i^v)_{v,i\in\mathcal{V},i\neq v}$ where the vectors t and y are binary.*

The proof of Theorem 3 is given in Appendix A. Theorem 3 implies that we can replace the follower's problem by its dual in model (17). Moreover, a direct byproduct of its proof is the following result:

Corollary 4 *Given a fixed attack $a = (a_v)_{v\in\mathcal{V}}$, there exists an optimal solution to* (18) *such that:*

1. *y and t are binary*
2. *$t_i^v = a_i, \quad i, v \in \mathcal{V}, i \neq v$*
3. *$t_i^i \geq a_i, \quad i \in \mathcal{V}$*
4. *$y_i^v \geq a_i, \quad i, v \in \mathcal{V}, i \neq v$*
5. *$t_v^v + y_i^v \leq 1 + a_i, \quad i, v \in \mathcal{V}, i \neq v.$*

Substituting t_i^v by a_i, renaming t_v^v as t_v, and adding the strengthening constraints from Corollary 4 to formulation (18) leads to the following reformulation of (17):

$$\text{minimize} \quad Z = MB + \sum_{i \in \mathcal{V}} (1 - a_i) t_i \tag{19a}$$

$$\text{subject to} \quad \sum_{v \in \mathcal{V}} a_v \leq K, \tag{19b}$$

$$B - \sum_{v \in \mathcal{V} \setminus \{i\}} y_i^v + t_i \geq 1 - (V - 1) a_i, \qquad i \in \mathcal{V} \tag{19c}$$

$$t_i \geq a_i, \qquad i \in \mathcal{V} \tag{19d}$$

$$y_i^v \geq a_i, \qquad i, v \in \mathcal{V}, i \neq v \tag{19e}$$

$$t_v + y_i^v \leq 1 + a_i, \qquad i, v \in \mathcal{V}, i \neq v \tag{19f}$$

$$t_v + y_j^v \geq 1, \qquad \{v, j\} \in \mathcal{E} \tag{19g}$$

$$y_i^v - y_j^v \leq a_i, \qquad \{i, j\} \in \mathcal{E}, v \in \mathcal{V} \setminus \{i, j\} \tag{19h}$$

$$y_j^v - y_i^v \leq a_j, \qquad \{i, j\} \in \mathcal{E}, v \in \mathcal{V} \setminus \{i, j\} \tag{19i}$$

$$y_i^v \in \mathbb{B}, \qquad i, v \in \mathcal{V}, i \neq v \tag{19j}$$

$$t_i^v \in \mathbb{B}, \qquad i, v \in \mathcal{V} \tag{19k}$$

$$a_v \in \mathbb{B}, \qquad v \in \mathcal{V} \tag{19l}$$

$$B \in \mathbb{R}_+. \tag{19m}$$

To solve (19) with a mixed-integer linear solver, we need to linearize the product $(1 - a_i)t_i$. As, by Corollary 4, $t_i \in \mathbb{B}$, we introduce auxiliary variables $l_i = (1 - a_i)t_i$, leading to the final mixed-integer linear formulation:

Formulation $\mathbb{F}4[K, M]$:

$$\text{minimize} \quad Z = MB + \sum_{i \in \mathcal{V}} l_i \tag{20a}$$

$$\text{subject to} \quad l_i \geq a_i - t_i, \qquad i \in \mathcal{V} \tag{20b}$$

$$l_i \in \mathbb{B}, \qquad i \in \mathcal{V} \tag{20c}$$

$$\text{(19b)} - \text{(19m)}. \tag{20d}$$

Finally, note that in fact Theorem 1 follows from Theorem 3. Indeed, in essence Theorem 3 states that formulation (17d)-(17j) is Totally Dual Integer, [12], which in turn implies Theorem 1 because a_v, $v \in \mathcal{V}$, are binary.

6 | COMPACT FORMULATIONS – EFFICIENCY AND EXTENSIONS

Below we discuss some selected issues related to the computation time efficiency of the NAOP formulations described in Section 5 and their possible extensions to other resilience measures.

TABLE 2 Lower bounds on the NAOP solutions resulting from linear relaxations (*coronet conus*, $K = 5, N[K] = 5$).

	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$
$\mathbb{F}1/\mathbb{F}2 : N = V - K,$ LB=M (gap)	1 (97%)	2 (96%)	3 (95%)	4 (94%)	5 (93%)
$\mathbb{F}1/\mathbb{F}2 : N = N[K],$ LB=14·M (gap)	14 (56%)	28 (53%)	42 (35%)	56 (18%)	70 (0%)
$\mathbb{F}3 :$ $N = V,$ LB calculated (gap)	2 (94%)	4 (93%)	6 (91%)	7 (90%)	9 (87%)
$\mathbb{F}4$ LB calculated (gap)	27 (16%)	46 (23%)	52 (20%)	55 (19%)	58 (17%)
$V[K, M]$ Optimal binary solution	32	60	65	68	70

6.1 | Linear relaxations, p-components defining constraints, number of variables

We start by noting that the minima of the linear relaxations (LR) of formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$ are the same, and hence these relaxations are of equal strength. This is because the sets of constraints (11b)-(11e) and (12b)-(12e) are the same in both cases, and the additional monotonicity constraint (11f) appearing in $\mathbb{F}1[K, M]$ has no effect on the minimum of LR. Therefore, when discussing the LR quality, we will only refer to $\mathbb{F}1[K, M]$.

The minimum of the objective function (11b) of the linear relaxation of $\mathbb{F}1[K, M]$, which is the absolute (independent of the network graph and valid for any K-node attack) lower bound (LB) on the total number of nodes in M largest p-components induced by a K-node attack in a V-node network can be calculated as follows. First, we note that for a given parameter N (where $N \leq V - K$), LB cannot be smaller than $\lceil M\frac{V-K}{N}\rceil$, the value obtained in the case when the set of all unaffected nodes is split into N components of equal (not necessarily integer) size. This is implied by constraints (11b)-(11d), which ensure that the total number of nodes in all p-components ($\sum_{n \in N} X^n$) is equal to $V - K$, and the observation that the sum of M largest p-components is minimized when all p-components are of equal size.

This minimum value of LB is achieved by the considered LR, since setting all a_v to $\frac{K}{V}$ (cf. (11b)) enables setting all x_v^n to $\frac{1-\frac{K}{V}}{N} = \frac{V-K}{V \cdot N}$ (cf. (11c)), which does not violate the common p-component constraints (11e). Then, according to (11d), X^n becomes $\frac{V-K}{N}$ for all n, and the sum of any M values X^n, i.e., the value of LB, becomes $\lceil M\frac{V-K}{N}\rceil$. Note that this formula is valid only for $M \leq N$ (for $M \geq N$, LB is equal to $V - K$).

Clearly, the quality of LB thus calculated increases as N decreases. To illustrate this consider the 75-node *coronet conus* network described in Section 7 and $K = 5$. Then $V - K = 70$, and for $N = V - K = 70$ (which is the maximum number of components over all network graph topologies), the size $\frac{V-K}{N}$ of each component is equal to $\frac{70}{70} = 1$. Hence, for any given M ($1 \leq M \leq N$), LB is equal to M, which corresponds to the case when the unaffected nodes are spilt into $V - K$ one-element components. The (trivial) lower bound LB=M obtained in this way means that for $N = V - K$ the linear relaxation of $\mathbb{F}1[K, M]$ is very weak.

However, when the network graph is known, we can compute the value of $N[K]$, i.e., the maximum number of components induced by a K-node attack (see formulation (16)), and use it for calculating LB. (Actually, the values $N = N[K]$ were used in all our computations performed with $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$.) For example, for $K = 5$ the maximum number of components in the considered network is equal to 5 ($N[5] = 5$, see Table 3) and $\frac{V-K}{N[K]} = \frac{70}{5} = 14$. Then, LB=$14 \cdot M$, which is a large improvement. Table 2 shows the so obtained lower bounds and compares them with the true minima of the objective function of NAOP (denoted by $V[K, M]$) taken from Table 9 for $K = 5$ and $M = 1, 2, \ldots, 5$. As could be expected, the gap between $V[K, M]$ and LB calculated for $N = N[K]$ decreases with M, and disappears for $M = 5$ (which must be the case since $N[5] = 5$).

The common p-component constraints (11e) and (12e) can be transformed (by adding $x^n_{\beta(e)}$ to both sides of the inequalities in (11e) and (12e), and using equalities from (11c) and (12c)) to the following equivalent (and much simpler) form:

$$x^n_{\alpha(e)} \le x^n_{\beta(e)} + a_{\beta(e)}, \qquad e \in \mathcal{E},\ n \in \mathcal{N}, \tag{21}$$

which simply means that when node $\beta(e)$ is not attacked ($a_{\beta(e)} = 0$) and does not belong to p-component n ($x^n_{\beta(e)} = 0$), then node $\alpha(e)$ (the other node of link e) cannot belong to p-component n either. This form of the considered constraints immediately shows that they are not violated by the above defined optimal solution of the linear relaxation (in which all x^n_v are equal). Clearly, using constraints (21) leads to sparser coefficient matrices in formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$, which can speed up processing the branch-and-bound nodes.

Finally, we note that in general $\mathbb{F}1[K, M]$ is more time consuming than $\mathbb{F}2[K, M]$. This phenomenon, apparently caused by the monotonicity constraint (11f), can be seen in the results of the numerical study.

Turning to formulation $\mathbb{F}3[K, M]$, we note that the common p-component constraints analogous to (21) can be used instead of constraints (15f):

$$x^n_{\alpha(e)} \le x^n_{\beta(e)} + a_{\beta(e)}, \qquad e \in \mathcal{E},\ n \in \mathcal{N},\ n \le \alpha(e). \tag{22}$$

Finding a closed LB formula for $\mathbb{F}3[K, M]$ is a difficult matter as it seems to depend on the network topology in a complicated way. We were able to find such formulae for two extreme cases: a fully disconnected network ($\mathcal{E} = \emptyset$) and a fully connected network ($\mathcal{E} = \mathcal{V}^{|2|}$). In the first case, LB=$\lceil M \frac{V-K}{V} \rceil$, and in the second case the value of LB can be efficiently calculated from V and K (the way to do this is omitted).

Anyhow, our numerical experiments reveal that the quality of the linear relaxations (both original and adjusted) of $\mathbb{F}3[K, M]$ (illustrated in Table 2) is lower as compared to $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$. An important reason for the weakness of the considered LB is the value of parameter N ($N = V$) that must be used and which is much larger than $N = N[K]$ used in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$ (as shown in Table 3). This has a negative effect on time efficiency of formulation $\mathbb{F}3[K, M]$.

Another cause of the decreased time efficiency of $\mathbb{F}3[K, M]$ is the number of binary variables x^n_v, $v \in \mathcal{V}, n \in \mathcal{N}$, that define the p-components. As already mentioned, this number is equal to $V \cdot N$ (where $N = N[K] \approx K$) in formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$, and to $\frac{V(V+1)}{2}$ in formulation $\mathbb{F}3[K, M]$ (where $N = V$). Thus, the number of variables in $\mathbb{F}3[K, M]$ can be much greater than in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$. For example, for the 75-node *coronet conus* network and 10-node attacks, we have $N[K] = 10$ (see Table 3), and hence the number of variables defining the p-components in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$ is equal to 750, while in $\mathbb{F}3[K, M]$ it is equal to 2775. As shown in the numerical study, this significant difference, together with the weaker LR, makes formulation $\mathbb{F}3[K, M]$ less time efficient than its counterparts $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$, although $\mathbb{F}3[K, M]$ directly addresses the problem of symmetrical solutions.

Finally, we also computed the lower bounds resulting from the linear relaxation of $\mathbb{F}4[K, M]$. These LBs are of better quality than LBs resulting from the other formulations for smaller values of M, i.e., when the number of surviving nodes is not close to $V - K$. Otherwise, formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$ with $N = N[K]$ are better in this regard. In any case, formulation $\mathbb{F}4[K, M]$ suffers from its much larger size compared to the other ones when the problem needs to be solved as a mixed-integer program.

6.2 | Decreasing the number of pseudo components N in $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$

Even the tight upper bound $N[K] = \max_{a \in \mathcal{A}[K]} C(a)$ on the number of components induced by K-node attacks (which is actually used as the value of parameter N, i.e., as the upper bound on the number of p-components in formulations $\mathbb{F}1[K, M]$ and $\mathbb{F}2[K, M]$) can lead to an excessive number (equal to $V \cdot N[K]$) of binary variables x_v^n, $v \in \mathcal{V}$, $n \in \mathcal{N}$, that define the p-components. To speed up the computations, in such cases we can take into account the values of parameter N that are smaller than $N[K]$, especially since in the formulations under consideration, reducing the number of p-components N improves, the quality of their linear relaxation, as shown in Section 6.1.

Clearly, when the assumed value of N is less than $N[K]$, it can happen that all optimal solutions of the formulations under consideration contain only proper p-components, which may lead to solutions a^* that overestimate the true values of the minima of the objective function obtained for $N = N[K]$. Yet, if the computation time is substantially decreased, such an overestimation can be acceptable. This observation, illustrated by the following example, is further discussed at the end of Section 7.3.

Example: Consider the *cost266* network composed of $V = 37$ nodes (described in Section 7). Assume the network is equipped with 3 controllers ($M = 3$) and exposed to 15-node attacks ($K = 15$). Figures 4 (and 5) show an optimal attack a^* obtained by means of formulation $\mathbb{F}2[15, 3]$ for $N = N[15] = 17$. The set of nodes attacked by a^*, $\mathcal{V}(a^*) = \{5, 7, 8, 13, 18, 23, 24, 25, 26, 27, 28, 29, 35, 36, 37\}$, together with the incident links, is depicted in Figures 4 in cyan. In the same figure, the non-empty p-components of the resulting N-family are shown, each in different color. Note that one of them is a non-proper p-component composed of nodes 6 and 33, and the rest are proper p-components. The sizes of the non-empty p-components of this family (there are 13 of them) listed in the non-increasing order are as follows: $3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1$, which means that the minimum of the objective function is equal to 8 (the total number of nodes in the three largest p-components). The proper N-family induced by the considered attack contains 14 (proper) p-components and is obtained when the non-proper p-component $\{6, 33\}$ is split into two proper p-components $\{6\}$ and $\{33\}$. Now, the ordered sequence of the p-components sizes is as follows: $3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1$, and, as before, the sum of the first three sizes is equal to 8.

The next two figures illustrate the case when the value of parameter N in $\mathbb{F}2[15, 3]$ is set to 4. The optimized attack a' (with $\mathcal{V}(a') = \{1,4,5,6,8,9,10,14,15,16,21,22,23,25,33\}$) and the p-components in its optimized N-family are shown in Figure 6. All four p-components of this family are non-proper; two of them (black and yellow nodes) are composed of 6 nodes each, and the other two (blue and pink nodes) are composed of 5 nodes each. Hence, the minimized objective value is equal to 17 and is substantially greater than the true minimum equal to 8. Yet, the family $\mathcal{V}(c)$, $c \in \mathcal{C}(a')$, illustrated in Figure 7, composed of 12 (proper) p-components of the (ordered) sizes $4, 4, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1$, shows that after attack a' optimized for $N = 4$ actually 10 nodes survive, which is a reasonable approximation.

Note that above considerations do not apply to formulations $\mathbb{F}3[K, M]$ (where N must be equal to V) and $\mathbb{F}4[K, M]$ (where N is not used at all).

6.3 | Reducing the set of nodes targeted by attacks

Another issue, which concerns all five formulations of NAOP considered in this paper, is the number of attacked nodes K which, as illustrated in Section 7, can substantially influence the computation time needed to solve NAOP. Certainly, the number of all K-node attacks for a V-node network can be very large. For example, for the *cost266* network (described in Section 7) with $V = 37$ nodes and $K = 4$, the number of all (4-node) attacks is equal to $\binom{37}{4} = 66,045$; for the *coronet conus* network (also described in Section 7) with $V = 75$ nodes, this number is equal to $\binom{75}{4} = 3,764,376$.

To alleviate this issue, we may use a simple rule to reduce the set of potential node locations that are subject to attacks. According to this rule, a node $v \in \mathcal{V}$ is not attacked ($a_v = 0$) if it is of degree 2 ($\deg(v) = 2$) and whose both neighbors, say nodes w and u, have in total at least 3 neighbors besides node v (i.e., $\deg(w) + \deg(u) - 2 \geq 3$). This is illustrated in Figure 8, where node v (in gray) is assumed not to be attacked, and its two neighbors (in blue) have in

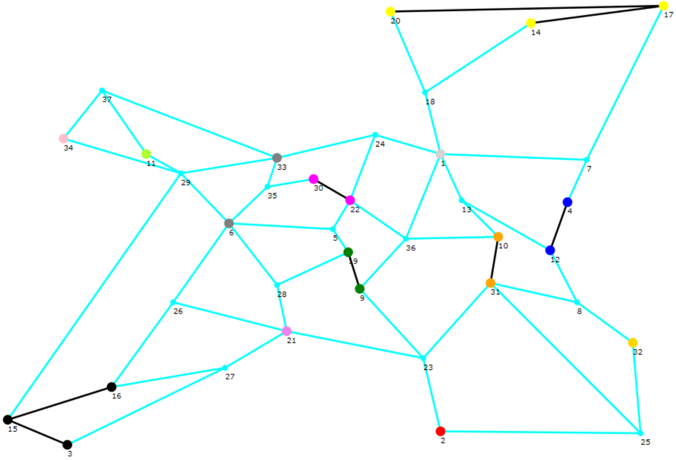

FIGURE 4 *cost266*: a non-proper *N*-family induced by attack a^* optimized for $N = 17$.

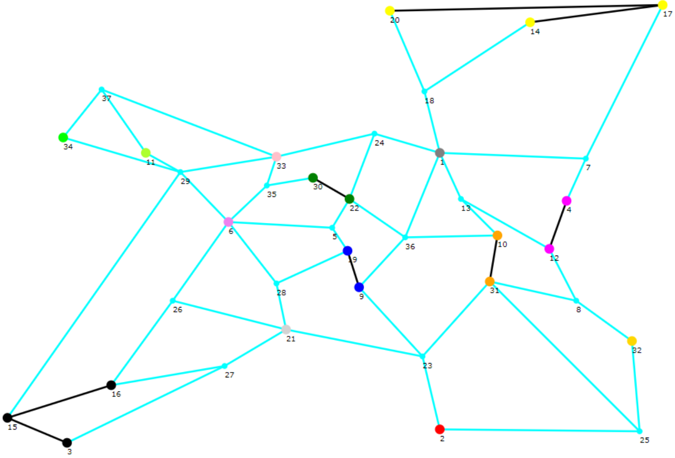

FIGURE 5 *cost266*: a proper family induced by attack a^* optimized for $N = 17$.

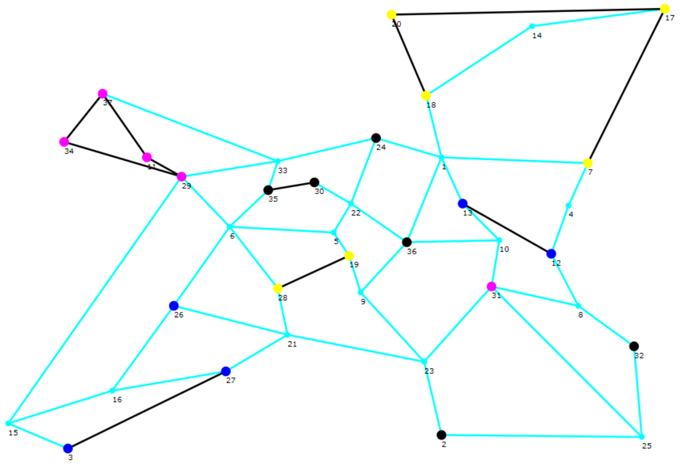

FIGURE 6 *cost266*: a non-proper N-family induced by attack a' optimized for $N = 4$.

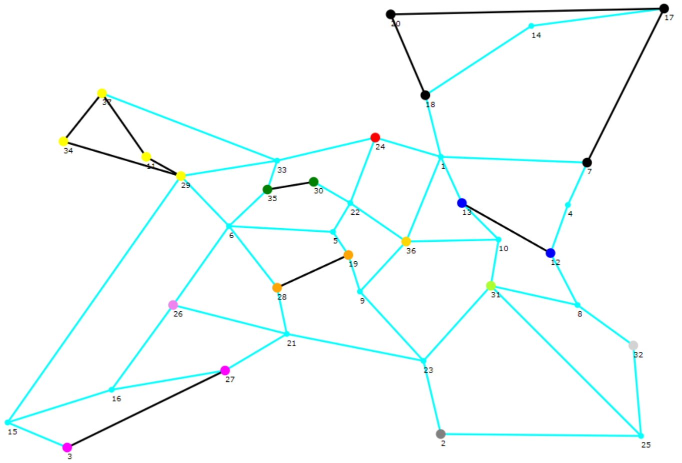

FIGURE 7 *cost266*: a proper N-family induced by attack a' optimized for $N = 4$.

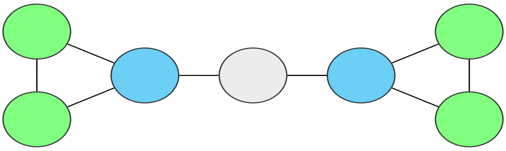

FIGURE 8 Elimination of potentially attacked nodes.

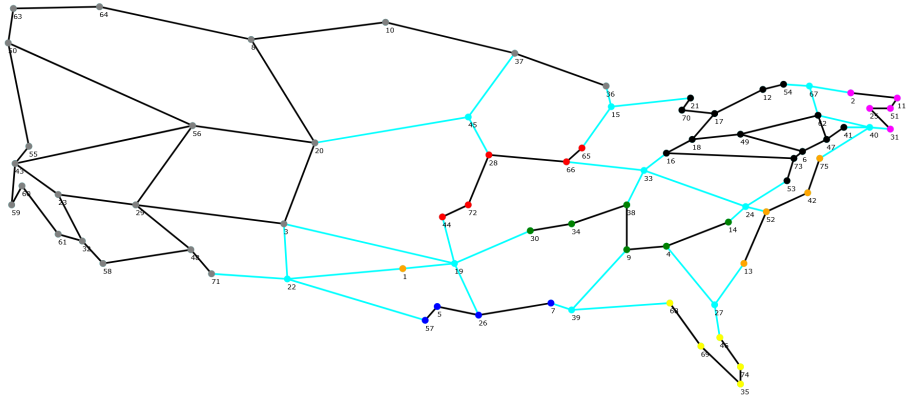

FIGURE 9 *coronet conus*: optimal attack for $M = 4, K = 10$.

total 4 neighbors (in green), not counting the grey node.

In fact, when applied to *cost266* the rule eliminates 9 (i.e., 24% of all nodes); these are nodes 2, 3, 4, 11, 14, 20, 30, 32, 34, see Figure 10 and Table 13. Hence, the number of 4-node attacks to be considered reduces to $\binom{28}{4} = 20,475$ (from $66,045$). For *coronet conus* this reduction eliminates as many as 32 nodes (i.e., 43% of all nodes); these are nodes 1, 2, 5, 7, 10, 12, 13, 14, 21, 30, 31, 34, 36, 41, 42, 44, 46, 53, 54, 55, 57, 58, 59, 61, 63, 64, 65, 68, 70, 71, 72, 75, see Figure 9 and Table 14. Now the number of 4-node attacks reduces to $\binom{43}{4} = 123,410$ (from $3,764,376$).

Certainly, it can happen that the considered rule eliminates nodes that are anecessary in optimal attacks. For example, for the network depicted in Figure 8, the gray node constitutes the unique optimal 1-node attack for $M = 1$. However, as illustrated in the numerical section, for the considered (meshed) network topologies this heuristic procedure can substantially reduce the computation times and at the same time in most cases deliver optimal solutions. Moreover, in the case when the obtained solutions turn out to be sub-optimal, they are very close to optimal.

6.4 | Extensions to other resilience measures

The resilience measure assumed in the node attack optimization problem (NAOP) considered in this paper is the number of surviving nodes $V(s, a)$ (see Section 3.1). Certainly, this particular measure, although important, is not the only one of interest.

For example, the measure used in this paper can be extended by introducing node weights $w(v)$, $v \in \mathcal{V}$, related to such features as importance, size, traffic load, etc., of a given node. This extension can be easily treated by the pseudo-component based formulations, simply by replacing equalities $X^n = \sum_{v \in \mathcal{V}} x_v^n$ with $X^n = \sum_{v \in \mathcal{V}} w(v) x_v^n$ in appropriate constraints.

Another important extension is to consider the total number of node-pairs in surviving components instead of the total number of surviving nodes. This measure can be incorporated into the pseudo-component based formulations by introducing a piece-wise linear function that counts the number of node-pairs in a set of X elements. This modification, introduced in [23], requires additional binary variables and constraints, and in general the resulting IP formulations are less time efficient than those used in this paper. However, developing reasonable bi-level formulations for both of the above extensions seems much more difficult.

Finally, note that the version of NAOP minimizing the total number of node-pairs in the surviving components for K-node attacks and $M = N[K]$ controllers is equivalent to a classic version of a well-known graph theory problem called Critical Node Detection (CND) (see [17, 1]). An efficient IP formulation for such a version of CND is described in [30].

7 | NUMERICAL STUDY

The paper documents our search for an effective method of solving the node attack optimization problem NOAP defined in Section 3.2. We focus on the process of method development and its results. We show how we explore two different solution approaches: the first one based on the direct optimization model using the graph-related properties of problem solutions, and the second one based on the bi-level optimization model of the considered problem. In our search for an effective problem solution method we were developing consecutive variants of the models, systematically analysing their performance on the selected data sets.

For method development we were using two well-known mesh network topologies that we had adopted in our previous research: a middle-size network and a large-size network. While developing the iterative solution method presented in our previous research we could solve the problem to optimality for a sample pair of K and M parameter values ($K = 4$ and $M = 6$) for the medium-size network but were unable to do that for the large-size network.

Now, for each of the two network topologies, we consider the problem instances corresponding to the values of K and M parameters ranging from 1 to 10. Thus, our data set consists of 200 instances and we use all these instances for each developed variant of the problem formulation (there are 8 of them). Such an approach seems to be preferable to using a larger number of network instances and a selected K and M value pairs, as while keeping the effort of running numerical experiments manageable, it also enables systematically analysing the dependence of the properties and the complexity of the developed models on the K and M parameter values, and provides for gaining more insight into problem characteristics.

The two networks studied in this paper are widely used in the literature; they are typical representatives of the mesh network topologies described in SNDlib [21], a library of test instances for survivable fixed telecommunication network design, available at http://sndlib.zib.de.

For the numerical study presented below, all applied IP formulations and algorithms were implemented as AMPL models and executed using the commercial software package AMPL/CPLEX 22.1 on a standard laptop.

7.1 | Description of the networks used in the study

The numerical results presented below illustrate the performance of the developed optimization models for two network instances, *cost266* and *coronet conus*. The first, a medium-size pan-European network (depicted in Figure 10) con-

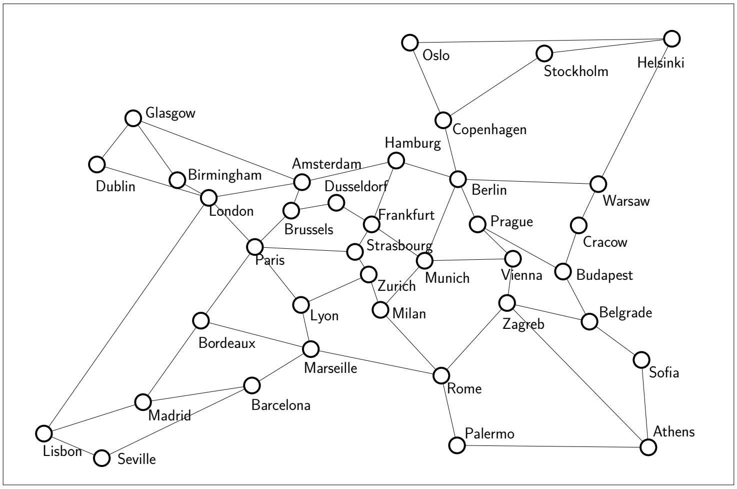

FIGURE 10 Topology of the *cost266* network.

sisting of $V = 37$ nodes and $E = 57$ links is described in SNDlib. The second, a large-size pan-American network (shown in Figure 11), composed of $V = 75$ nodes and $E = 99$ links is described at http://www.monarchna.com/topology.html (see also [31]). The numbers assigned to node locations in the considered networks are listed in Tables 13 and 14 in Appendix B; these numbers are used to identify the node locations when presenting the numerical results.

We end the description of the example networks with Table 3, where the maximum numbers of connected components $N[K] = \max_{a \in \mathcal{A}[K]} C(a)$ induced by K-node attacks are shown. Note that these numbers, basic parameters for formulations $\mathbb{F}1[K,M]$ and $\mathbb{F}2[K,M]$, were computed by means of formulation (16) in a very short time, equal to fractions of a second for *cost266*, and less than 5 seconds for *coronet conus*. Because of that, and because each value $N[K]$ is computed only once for each $K = 1, 2, \ldots, 10$ (i.e., the range of parameter K considered in the study), we do not include the time spent on computing these values in the computation times reported in Sections 7.2 and 7.3.

TABLE 3 Maximum number of connected components for attacks of size K.

$N[K]$/time [s]	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$	$K = 10$
cost266	1 0.15	3 0.08	3 0.17	5 0.18	5 0.30	6 0.36	7 0.39	8 0.28	9 0.42	10 0.33
coronet conus	1 0.35	2 0.53	3 1.33	4 3.78	5 4.27	6 3.34	8 4.28	9 4.75	11 4.14	12 3.02

7.2 | Results for *cost266*

The results of applying our optimization formulations to the *cost266* network are shown in Tables 4-8 below and in Tables 15-20 in Section C.1 of Appendix C. In each case, the results were obtained for the range of parameters $K = 1, 2, \ldots, 10$, and $M = 1, 2, \ldots, 10$. For simplicity, in the following discussion we write $\mathbb{F}1, \mathbb{F}2, \mathbb{F}3, \mathbb{F}4$ instead of $\mathbb{F}1[K,M]$, $\mathbb{F}2[K,M]$, $\mathbb{F}3[K,M]$, $\mathbb{F}4[K,M]$.

In Table 4, the exact optimal values $V(s(a^*), a^*)$ (denoted by $V[K,M]$ for each pair of parameters K, M), defined

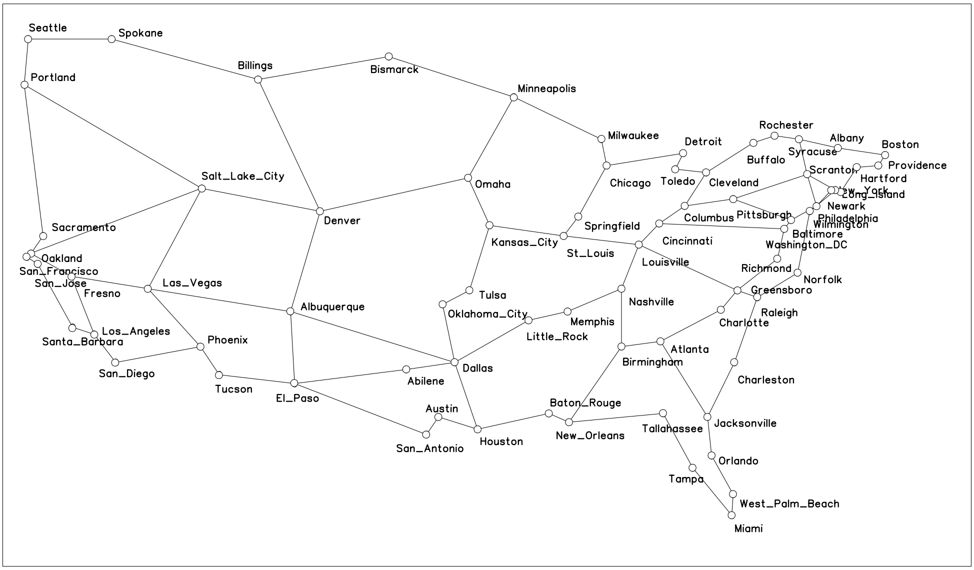

FIGURE 11 Topology of the *coronet conus* network.

in (6), are presented (all these values were obtained by each formulation $\mathbb{F}2$, $\mathbb{F}3$, $\mathbb{F}4$ within the time limit of 300 seconds). Additionally, in the last column, the upper bounds (UB[K]=$V - K$) on $V(s(a^*), a^*)$ are shown for all K. As expected, the values of $V[K, M]$ are decreasing with K and non-decreasing with M. The table also confirms the fact that for a fixed K, the upper bound on the guaranteed number of surviving nodes, i.e., UB[K]= $V - K$, is reached for the first time for $M = N[K]$. Note that for the network under consideration $N[K] = K$ for all K except for $K = 3, 5$, for which $N[K] = K + 1$. Another observation is that attacks with large K ($K \geq 9$) are rather devastating for small M ($M \leq 4$).

In Tables 5 and 15-16, computation times for compact formulations $\mathbb{F}2$-$\mathbb{F}4$ are shown. The main observation is that for these three formulations the computation times are quite short, less (usually much less) than 100 seconds in all cases. In general, with a few exceptions, $\mathbb{F}2$ is the fastest and $\mathbb{F}4$ is visibly faster than $\mathbb{F}3$.

Tables 6 and 17-18 show the computation times for $\mathbb{F}2$-$\mathbb{F}4$ run for the sets of potentially attacked nodes reduced according to the rule described in Section 6.3, i.e., for formulations $\mathbb{F}2R$-$\mathbb{F}4R$. It is clear that such a reduction leads to substantially shorter computation times for each formulation and these times are almost negligible. Most importantly, for all pairs K, M under consideration, the so obtained optimal values $V[K, M]$ are exact (i.e., equal to those shown in Table 4) which means that application of this (heuristic) reduction rule can be effective.

Finally, Tables 19-20 reveal that computation times for $\mathbb{F}1$ and $\mathbb{F}1R$ can be much longer than for the remaining compact formulations, especially when $K \geq 8$, where the assumed time limit (set to 300 seconds) was frequently reached. Certainly, when the timeout did not occur, the optimal solutions were found with both $\mathbb{F}1$ and $\mathbb{F}1R$. Interestingly, in such cases the best solutions (called incumbents) found by $\mathbb{F}1$ before the timeout occurred were in all cases optimal, besides the case of $K = 10, M = 8$, for which $V[10, 8] = 23$ instead of 22 was obtained; for $\mathbb{F}1R$ all such incumbents were optimal. Moreover, these incumbents were found quite quickly, which means that the solver spent most of the time trying to prove optimality of the considered incumbents. In fact, this phenomenon was common in the runs reported in Tables 5-6 and 15-20.

To end this section, let us discuss the results for the min-max algorithm CPGA presented in Section 4.2. In Table 7 the values of $V[K, M]$ computed by the algorithm are shown. For any given pair K, M, the corresponding entry shows the value of $V[K, M]$ (before the slash) and the number of iterations, i.e., the number of generated placements, needed to achieve the optimal solution (after the slash). In the computations, the limit on the number of iterations performed by the algorithm was set to 100. In the table, the entries with the dash correspond to the K, M cases for which the time limit of 4 hours was reached before 100 iterations were executed. In the remaining cases (for which the results is shown), the correct optimal values $V[K, M]$ were obtained, except for $K = 5, M = 5$: in this particular case the limit of 100 iterations was reached within the time limit with the incorrect value $V[K, M] = 31$, instead of the optimal value equal to 32. This is because if CPGA is terminated before the stopping criterion given in Step 1 of CPGA (see Section 4.2) is fulfilled, the final value of $V[K, M]$ is in general less than the optimal one since not all placements in $\mathcal{S}[M]$ are taken into account in evaluating $\max_{s \in \mathcal{S}} V(s, a^*)$. This means that in such cases CPGA overestimates the strength of the attack.

Table 8 reveals that computation times for CPGA (spent mostly in Step 2 solving $\mathbb{A}[K, \mathcal{S}]$) are much longer than for the compact formulations, and become unacceptable for $K \geq 5$ and $M \geq 5$. Of course, this effect is more pronounced for large networks, and therefore we do not discuss the results of CPGA for *coronet conus* in Section 7.3.

TABLE 4 *cost266*: number of surviving nodes for the optimal attack when the best placement is applied exact results obtained with $\mathbb{F}2$, $\mathbb{F}3$, $\mathbb{F}4$

$V[K,M]$	$M=1$	$M=2$	$M=3$	$M=4$	$M=5$	$M=6$	$M=7$	$M=8$	$M=9$	$M=10$	UB[K]
$K=1$	36	36	36	36	36	36	36	36	36	36	36
$K=2$	29	34	35	35	35	35	35	35	35	35	35
$K=3$	19	32	34	34	34	34	34	34	34	34	34
$K=4$	17	27	31	32	33	33	33	33	33	33	33
$K=5$	13	25	29	31	32	32	32	32	32	32	32
$K=6$	13	20	26	29	30	31	31	31	31	31	31
$K=7$	8	16	22	27	28	29	30	30	30	30	30
$K=8$	7	13	18	23	26	27	28	29	29	29	29
$K=9$	6	11	15	19	22	25	26	27	28	28	28
$K=10$	5	9	13	16	19	22	24	25	26	27	27

TABLE 5 *cost266*: computation times for $\mathbb{F}2$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$K = 2$	0.07	0.09	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 3$	0.13	0.14	0.02	0.01	0.01	0.01	0.02	0.03	0.01	0.01
$K = 4$	0.44	0.52	0.52	0.55	0.01	0.01	0.02	0.02	0.01	0.02
$K = 5$	0.59	0.53	0.67	0.62	0.01	0.02	0.01	0.02	0.01	0.01
$K = 6$	0.83	0.90	1.05	1.10	1.03	0.02	0.02	0.01	0.01	0.02
$K = 7$	0.98	1.22	1.49	2.04	2.27	1.28	0.03	0.03	0.03	0.02
$K = 8$	1.52	1.88	2.56	2.85	7.41	9.07	2.92	0.02	0.02	0.02
$K = 9$	2.52	3.00	2.53	7.24	7.16	9.48	8.50	3.35	0.03	0.02
$K = 10$	3.86	9.85	3.80	11.4	13.0	16.12	14.2	14.6	27.8	0.03

TABLE 6 *cost266*: computation times for $\mathbb{F}2R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
$K = 2$	0.07	0.09	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 3$	0.12	0.12	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 4$	0.48	0.42	0.41	0.40	0.02	0.01	0.01	0.02	0.01	0.02
$K = 5$	0.38	0.46	0.49	0.45	0.01	0.01	0.01	0.01	0.01	0.02
$K = 6$	0.75	0.61	0.89	0.86	0.92	0.01	0.04	0.01	0.01	0.01
$K = 7$	0.81	1.29	1.57	1.99	1.58	1.71	0.02	0.03	0.01	0.01
$K = 8$	1.13	1.85	1.95	7.08	7.12	6.79	2.28	0.02	0.02	0.02
$K = 9$	2.79	2.95	7.19	8.26	9.28	9.57	7.37	2.74	0.03	0.02
$K = 10$	9.55	2.80	13.63	11.16	16.95	10.34	11.55	14.72	4.31	0.03

7.3 | Results for *coronet conus*

The numerical results for the *coronet conus* network are shown in Tables 9-11 below and in Tables 21-24 in Section C.2 of Appendix C. Table 9 (analogous to Table 4) shows the values of $V[K, M]$ obtained for *coronet conus* with $\mathbb{F}2$ and $\mathbb{F}2R$. More specifically, a single $V[K, M]$ value in a given table entry indicates that this value was obtained with both formulations; if there are two values, the first value was obtained with $\mathbb{F}2$ and the second value (in parentheses) with $\mathbb{F}2R$. Note that for all $K \leq 8$ the $V[K, M]$ values obtained with $\mathbb{F}2$ are optimal since they were achieved within the time limit of 600 seconds (see Table 10). The solutions of $\mathbb{F}2R$ (also obtained within the time limit) are also optimal, except for $V[5, 1]$ – in this case the value obtained with $\mathbb{F}2R$ is equal to 33. For $K = 9, 10$, the same observations are valid for the cases with $V[K, M]$ printed in black, i.e., when computations with $\mathbb{F}2$ stopped successfully before the time limit elapsed (see Table 10). However, in the cases with $V[K, M]$ printed in red, i.e., when computations with $\mathbb{F}2$ were stopped because of the timeout (see Table 10), we cannot conclude that the obtained values are optimal. In fact, for $K = 9$, $M = 6, 8, 9$, and $K = 10$, $M = 7, 9$, better results (but also potentially suboptimal) are obtained with $\mathbb{F}2R$. (Note that all computations with $\mathbb{F}2R$ reported in Table 9 were stopped successfully, see Table 11.)

Given the size of the *coronet conus* network, the computation times for $\mathbb{F}2$ (shown in Table 10) are acceptable: the $V[K, M]$ values of the best incumbent, which are shown in Table 9 for the timeout cases occurring with $K = 9, 10$, are most likely not far from optimal. The computation times for $\mathbb{F}2R$ (shown in Table 11) are very good (no timeouts). Taking into account that $V[K, M]$ values obtained with $\mathbb{F}2R$ are very close to optimal (and very close, or even better, to those obtained with $\mathbb{F}2$ until a timeout occurred), we observe that using the reduction rule introduced in Section 6.3 is advantageous also for *coronet conus*.

Computation times for $\mathbb{F}3, \mathbb{F}4, \mathbb{F}3R, \mathbb{F}4R$ are shown in Tables 21-24. In these tables the results for $K = 9, 10$ are skipped, as for almost all cases of M the timeout of 600 seconds occurred. For the considered cases (with $K \leq 8$) it is seen that computation times for $\mathbb{F}3$ are apparently longer than for $\mathbb{F}2$, and for $\mathbb{F}4$ they become even worse, as in most the timeout was encountered. For $\mathbb{F}3R$, in turn, the computation times become acceptable but still apparently longer than for $\mathbb{F}2R$. For $\mathbb{F}4R$, however, the computation times are much worse than for $\mathbb{F}2R$.

Importantly, in all cases considered (including timeout cases), the final incumbents were found quite quickly and almost all of them were of good quality.

Finally, it should be noted that we do not present here the results for $\mathbb{F}1, \mathbb{F}1R$ and CPGA, because for all of them the computation times (except for small values of K) turned out to be unacceptable.

We end this section with illustrating the effect (discussed in Section 6.2) of decreasing the value of parameter N (below its correct value $N[K]$) on the efficiency of formulation $\mathbb{F}2$ for the case $K = 9, M = 5$ (where $N[K] = 11$). The results of this experiment for $N = 11, 10, \ldots, 5$, are summarized in Table 12. These are: computation times for executing $\mathbb{F}2$ (first row), values of $V[9, N]$ optimized with $\mathbb{F}2$ (second row), sizes of optimal p-components in non-increasing order (third row), $V'[9, N]$ – values of $V[9, N]$ calculated for the proper family of components $\mathcal{F}(a^*, N) = C(a^*)$ resulting from the optimized attack a^* for a given N (fourth row), sizes of the components in $\mathcal{F}(a^*, N) = C(a^*)$ in non-increasing order (fifth row).

As indicated in Table 12, the correct optimal result $V[9, 5] = 55$ (for $N = N[9] = 11$) was found by $\mathbb{F}2$ in 525 seconds. The optimized attack a^* is composed of the set of nodes $\mathcal{V}(a^*) = \{17, 19, 22, 24, 27, 30, 33, 40, 67\}$ and splits the set of the remaining 66 nodes into $C(a^*) = 9$ (proper) components, see Figure 12. The sizes of these components in non-increasing order are shown in the last column of Table 12 in row "sizes (pseudo)" and, because all the components are proper, also in row "sizes (proper)". When the value of parameter N decreases from 11 to 9, nothing, except the computation time which decreases approximately two times, is changed because the same optimal solution is found. In fact, attack a^* is found by $\mathbb{F}2$ also for $N = 8, 7, 6$. In consequence, for $N = 8$ one non-proper p-component appears

TABLE 7 *cost266*: number of surviving nodes for the optimal attack when the best placement is applied results obtained with the CPGA algorithm

$V[K,M]$/iter	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$	UB[K]
$K = 1$	36/2	36/2	36/2	36/2	36/2	36/1	36/1	36/1	36/1	36/1	36
$K = 2$	29/3	35/8	35/8	35/8	35/8	35/5	35/4	35/4	35/4	35/4	35
$K = 3$	19/4	32/10	34/27	34/25	34/24	34/15	34/13	34/14	34/10	34/12	34
$K = 4$	17/5	27/8	31/24	32/20	33/45	33/41	33/30	33/29	33/26	33/28	33
$K = 5$	13/6	25/11	29/32	31/52	31^*/100	–	–	–	–	–	32
$K = 6$	13/7	20/12	26/26	29/51	–	–	–	–	–	–	31
$K = 7$	8/8	16/14	22/27	27/45	–	–	–	–	–	–	30
$K = 8$	7/7	13/13	18/17	23/25	–	–	–	–	–	–	29
$K = 9$	6/11	11/15	15/18	19/19	–	–	–	–	–	–	28
$K = 10$	5/13	9/21	13/23	16/24	–	–	–	–	–	–	27

TABLE 8 *cost266*: computation times for the CPGA algorithm

time	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	< 1s	< 1s	< 1s	< 1s	< 1s	< 1s	< 1s	< 1s	< 1s	< 1s
$K = 2$	2s	2s	8s	8s	7s	5s	4s	4s	4s	4s
$K = 3$	1s	15	150s	151s	108s	16s	32	20	10	28
$K = 4$	3s	9s	177s	165s	19m10s	12m56s	6m10s	5m3s	3m54s	2m59s
$K = 5$	4s	30s	383s	54m52s	3h31m8s	–	–	–	–	–
$K = 6$	5s	24s	172s	59m9s	–	–	–	–	–	–
$K = 7$	4s	32s	92s	1h5m6s	–	–	–	–	–	–
$K = 8$	6s	33s	85s	9m20s	–	–	–	–	–	–
$K = 9$	10s	25s	89s	153s	–	–	–	–	–	–
$K = 10$	15s	60s	127s	157s	–	–	–	–	–	–

TABLE 9 *coronet conus*: number of surviving nodes for the optimal attack when the best placement is applied results obtained with $\mathbb{F}2$ ($\mathbb{F}2R$)

$V[K,M]$	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$	UB[K]
$K = 1$	74	74	74	74	74	74	74	74	74	74	74
$K = 2$	68	73	73	73	73	73	73	73	73	73	73
$K = 3$	64	70	72	72	72	72	72	72	72	72	72
$K = 4$	36	66	69	71	71	71	71	71	71	71	71
$K = 5$	32 (33)	60	65	68	70	70	70	70	70	70	70
$K = 6$	27	48	62	65	67	69	69	69	69	69	69
$K = 7$	21	42	57	62	65	66	67	68	68	68	68
$K = 8$	18	35	51	56	61	64	65	66	67	67	67
$K = 9$	16 (17)	32 (33)	45	50	55	61 (59)	63	65 (64)	66 (65)	66	66
$K = 10$	14 (15)	29	41	46	51	56	60 (59)	61	64 (63)	63 (65)	65

TABLE 10 *coronet conus*: computation times for $\mathbb{F}2$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$K = 2$	0.12	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 3$	0.43	0.45	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
$K = 4$	0.66	0.95	0.98	0.02	0.03	0.02	0.02	0.02	0.02	0.01
$K = 5$	1.60	2.67	3.25	3.38	0.02	0.02	0.02	0.02	0.03	0.02
$K = 6$	4.28	5.23	19.3	20.3	21.9	0.02	0.03	0.03	0.03	0.03
$K = 7$	12.2	37.4	80.3	84.0	99.8	296	199	0.03	0.03	0.04
$K = 8$	53.3	59.9	113	163	253	298	473	305	0.04	0.04
$K = 9$	111	157	387	318	525	600	600	600	600	600
$K = 10$	447	600	600	600	600	600	600	600	600	600

TABLE 11 *coronet conus*: computation times for $\mathbb{F}2R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
$K = 2$	0.18	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 3$	0.27	0.26	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 4$	0.56	0.77	0.72	0.02	0.02	0.02	0.02	0.02	0.02	0.02
$K = 5$	1.25	1.71	2.42	1.97	0.03	0.04	0.03	0.02	0.03	0.02
$K = 6$	2.44	2.91	6.46	14.3	16.0	0.04	0.05	0.02	0.03	0.03
$K = 7$	24.1	29.2	49.5	40.9	33.3	61.5	153	0.06	0.04	0.04
$K = 8$	58.9	48.9	91.9	99.6	165	197	200	201	0.04	0.05
$K = 9$	81.0	72.6	74.0	76.4	233	333	464	399	341	0.06
$K = 10$	198	265	392	270	354	372	397	322	299	0.06

in the optimized N-family of p-components (as a result of merging the last two proper p-components) but this does not affect the optimal value od the objective function. In the next two cases $N = 7, 6$, there are 2 and 3 non-proper p-components, respectively, the minimum of the objective function grows to 56 and 60, and the computation time decreases. However, since the optimal attack a^* was found in all these cases, using its proper family $\mathcal{F}(a^*, N)$ instead of the obtained N-family leads to the correct value of the minimum of the objective function, i.e., $V'[9, 5] = 55$. Finally, note that the case $N = 5$ is different. Now, because all five potential p-components can be equipped with one of $M = 5$ controllers, a trivial attack a with $\mathcal{V}(a) = \{1, 2, \ldots, 9\}$ and one large proper component composed of the remaining 66 nodes is obtained with $\mathbb{F}2$. Hence, using the proper family $\mathcal{F}(a, N)$ does not help in this case.

In summary, decreasing the value of parameter N can be useful, as it may find an optimal solution much faster than for $N = N[K]$. In the considered case the decrease in computation time is quite impressive – from 525 to 38 seconds. Note that even if the obtained values of $V'[K, M]$ are not optimal, they can be used as initial upper bounds in the linear solver.

TABLE 12 Effects of decreasing N for $\mathbb{F}2$ (*coronet conus*, $K = 9$, $M = 5$)

	$N = 5$	$N = 6$	$N = 7$	$N = 8$	$N = 9$	$N = 10$	$N = 11$
time [s]	0	38	76	112	276	255	525
$V[9, 5]$	66	60	56	55	55	55	55
sizes (pseudo)	66 66	30, 9, 9, 6, 6, 6	30, 9, 6, 6, 5, 5, 5	30, 9, 6, 5, 5 4, 4, 3	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1
$V'[9, 5]$	66	55	55	55	55	55	55
sizes (proper)	66 66	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1	30, 9, 6, 5, 5, 4, 4, 2, 1

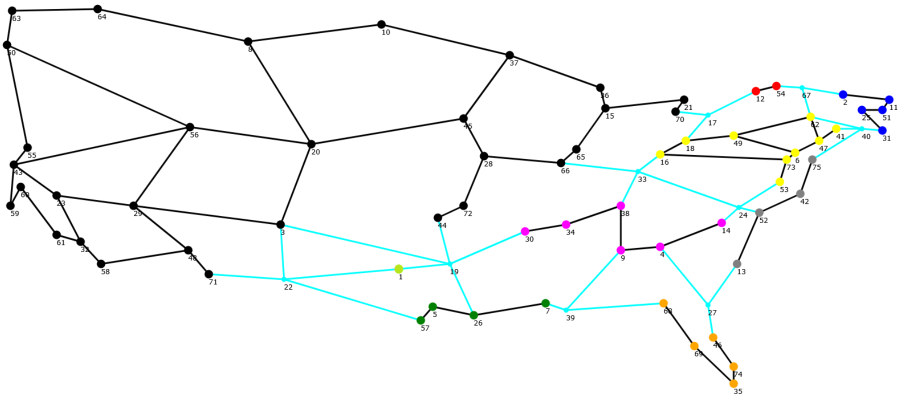

FIGURE 12 Effects of decreasing N for F2 (*coronet conus*, $K = 9$, $M = 5$).

7.4 | Comments

Overall, the numerical results presented above show that approaching the Node Attack Optimization Problem (NAOP) with compact formulations provides effective means for finding exact solutions in acceptable computation time, and that this approach clearly outperforms the alternative approach based on iterative solving of a non-compact NAOP formulation.

In the case of the 37-node *cost266* network, formulations $\mathbb{F}2$, $\mathbb{F}3$, $\mathbb{F}4$ perform very well and deliver optimal solutions very quickly, in many cases in less than 1 second. Formulation $\mathbb{F}1$, characterized by a straightforward and easy-to-understand form, is generally less efficient than other compact formulations, but for small values of K (say $K \leq 7$) it performs just as well.

Another important observation is that for *cost266* the procedure of reducing the number of locations exposed to potential attacks according to the node degree rule is beneficial. This procedure visibly decreases computation times for all formulations, at the same time preserving optimality of the solutions.

In the case of the 75-node *coronet conus* network, computation times generally increase in all cases, but the approach based on compact formulations is still effective taking into account the large size of *coronet conus* and the wide range of K and M. The best performance is observed for $\mathbb{F}2R$, i.e., the version of $\mathbb{F}2$ with the reduced set of nodes that are exposed to attacks, which was able to deliver optimal solutions before the timeout and in many cases in less than a few seconds. In fact, combining the results of $\mathbb{F}2$ and $\mathbb{F}2R$ makes it possible to find optimal solutions for all $K \leq 8$ and very good suboptimal solutions for $K = 9, 10$. It should also be noted that for $\mathbb{F}1$ and $\mathbb{F}2$ reducing the assumed number of components N can lead to a significant reduction in computation times at the expense of (possibly) slight underestimation of the strength of the optimized attack.

Finally, let us note that a nice feature of the introduced compact formulations is that in most cases incumbents (integer solutions) of good quality are obtained in a relatively short computation time.

8 | FINAL REMARKS

Resilience of service networks to node-targeted attacks is an important topic. Two complementary problems can be defined. From the network operator's point of view, the problem consists in optimally placing controllers at nodes to maximize the number of service nodes surviving a set of potential node attacks. From the attacker's point of view, the objective is to select the attack that minimizes the number of surviving service nodes given a set of potential controllers placements. In this paper, we focus on the node attack optimization problem, seen in the context of game theory. Two major new contributions are presented: compact formulations of the problem based on the notion of pseudo-components, and a formulation based on a bilevel model. This last model is tackled through a reformulation as a mixed-integer program based on important properties of optimal solutions of the controllers placement problem and its dual.

Numerical experiments performed on two realistic mesh network topologies widely used in the literature show that both approaches are competitive. Nevertheless, the linear relaxations of the proposed models are still quite weak, and deriving specific strong valid inequalities is an avenue that should be explored in the future.

A challenging problem also left out for future research is to tackle the problem from the network operator's perspective, i.e., to find the best controllers placement given that the attacker will respond with its best possible node attack. To the best of our knowledge, the difficulty is the lack of structural results for the node attack (lower-level) problem that would allow for an effective bilevel-based reformulation of the problem as a mixed-integer program.

References

[1] B. Addis, M. Di Summa, and A. Grosso, *Identifying critical nodes in undirected graphs: Compexity results and polynomial algorithms for the case of bounded treewidth*, Discr. Appl. Math. **161** (2013), 2349–2360.

[2] E. Calle, S.G. Cosgaya, D. Martínez, and M. Pióro, *Solving the Backup Controller Placement Problem in SDN under simultaneous targeted attacks*, 11th International Workshop on Resilient Networks Design and Modeling (RNDM), 2019, pp. 1–7.

[3] E. Calle, D. Martínez, M. Mycek, and M. Pióro, *Resilient backup controller placement in distributed SDN under critical targeted attacks*, Int. J. Critical Infrastructure Protection **33** (2021), 12–260.

[4] M. Campêlo, V.A. Campos, and R.C. Corrêa, *On the asymmetric representatives formulation for the vertex coloring problem*, Discr. Appl. Math. **156** (2008), 1097–1111.

[5] P. Carroll, B. Fortz, M. Labbé, and S. McGarraghy, *A branch-and-cut algorithm for the ring spur assignment problem*, Networks **61** (2013), 89–103.

[6] B. Colson, P. Marcotte, and G. Savard, *Bilevel programming: A survey*, 4OR **3** (2005), 87–107.

[7] T. Das, V. Sridharan, and M. Gurusamy, *A survey on Controller Placement problem in SDN*, IEEE Commun. Surveys Tutorials **22** (2020), 472–503.

[8] A. de Sousa, D. Mehta, and D. Santos, *The Robust Node Selection Problem aiming to Minimize the Connectivity Impact of any* p *Node Failures*, Design of Reliable Communication Networks (DRCN), 2017, pp. 138–145.

[9] S. Dempe, *Foundations of bilevel programming*, Springer Science & Business Media, 2002.

[10] W. Ellens, *Effective resistance and other graph measures for network robustness*, Master's thesis, Leiden University, 2011.

[11] L. Faramondi, R. Setola, S. Panzieri, F. Pascucci, and G. Oliva, *Finding critical nodes in infrastructure networks*, Int. J. Critical Infrastructure Protection **20** (2017), 3–15.

[12] F. Giles and W. Pulleyblank, *Total dual integrality and integer polyhedra*, Linear Algebra its Appl. **25** (1979), 191–196.

[13] B. Heller, R. Sherwood, and N. McKeown, *The controller placement problem*, 1st workshop on Hot Topics in Software Defined Networks (HotSDN), 2012, pp. 7–12.

[14] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, *Multi-controller based software-defined networking: A survey*, IEEE Access **6** (2018), 15980–15996.

[15] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, *Attack robustness and centrality of complex networks*, PLOS ONE **8** (2013).

[16] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt, *A survey on mixed-integer programming techniques in bilevel optimization*, EURO J. Comput. Optim. **9** (2021), 100007.

[17] M. Lalou, T. Mohammed Amin, and H. Kheddouci, *The Critical Node Detection Problem in networks: A survey*, Comput. Sci. Review **28** (2018), 92–117.

[18] J. Lu, Z. Zhang, T. Hu, P. Yi, and J. Lan, *A survey of controller placement problem in software-defined networking*, IEEE Access **7** (2019), 24290–24307.

[19] M. Manzano, J. Marzo, E. Calle, and A. Manolovay, *Robustness analysis of real network topologies under multiple failure scenarios*, 17th European Conference on Networks and Optical Communications (EuCNC), 2012, pp. 1–6.

[20] M. Mycek, M. Pióro, A. Tomaszewski, and A. de Sousa, *Optimizing primary and backup SDN controllers' placement resilient to node-targeted attacks*, 17th International Conference on Network and Service Management (CNSM 2021), 2021.

[21] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, *SNDlib 1.0 – Survivable network design library*, Networks: An Int. J. **55** (2010), 276–286.

[22] N. Perrot and T. Reynaud, *Optimal placement of controllers in a resilient SDN architecture*, 12th International Conference on the Design of Reliable Communication Networks (DRCN), 2016, pp. 145–151.

[23] M. Pióro, M. Mycek, and A. Tomaszewski, *Network protection against node attacks based on probabilistic availability measures*, IEEE Trans. Network Service Manage. **18** (2021), 2742–2763.

[24] M. Pióro, M. Mycek, A. Tomaszewski, and A. de Sousa, *On joint primary and backup controllers placement optimization against node-targeted attacks*, 12th International Workshop on Resilient Networks Design and Modelling (RNDM 2022), 2022.

[25] D.F. Rueda, E. Calle, and J.L. Marzo, *Improving the robustness to targeted attacks in software defined networks (SDN)*, 13th International Conference on Design of Reliable Communication Networks (DRCN 2017), 2017, pp. 1–8.

[26] D.F. Rueda, E. Calle, and J.L. Marzo, *Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements*, J. Network Syst. Manage. **25** (2017), 269–289.

[27] D. Santos, A. de Sousa, and C.M. Machuca, *Combined control and data plane robustness of SDN networks against malicious node attacks*, 14th International Conference on Network and Service Management (CNSM), 2018, pp. 54–62.

[28] D. Santos, A. de Sousa, and C.M. Machuca, *Robust SDN controller placement to malicious node attacks*, 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 2018, pp. 1–8.

[29] D. Santos, A. de Sousa, and C.M. Machuca, *The controller placement problem for robust SDNs against malicious node attacks considering the control plane with and without split-brain*, Ann. Telecommunications **74** (2019), 575–591.

[30] D. Santos, A. de Sousa, and P. Monteiro, *Compact Models for Critical Node Detection in Telecommunication Networks*, Electron. Notes Discr. Math. **64** (2018), 325–334.

[31] J.M. Simmons, *Optical Network Design and Planning (2nd edition)*, Springer, Switzerland, 2014.

[32] A.K. Singh and S. Srivastava, *A survey and classification of controller placement problem in SDN*, Int. J. Network Manage. **28** (2018).

[33] A. Tomaszewski, M. Pióro, and M. Mycek, *Min-Max Optimization of Attacks on Nodes vs. Max-Min Optimization of Controller Placements in Service Networks*, 10th International Network Optimization Conference (INOC 2022), 2022.

A | PROOFS OF THEOREM 1 AND THEOREM 3

Theorem 1 Given a fixed attack $a = (a_v)_{v \in \mathcal{V}}$, the linear relaxation of the MIP formulation (17d)-(17j) of the follower's problem admits an optimal solution $s = (s_v)_{v \in \mathcal{V}}$ where $s_v \in \mathbb{B}$ for all $v \in \mathcal{V}$.

Proof: Let $s = (s_v)_{v \in \mathcal{V}}$ be a fractional optimal solution of problem (17d)-(17j) for attack a. Let $\Phi(s, c) = \sum_{v \in \mathcal{V}(c)} \phi(s, v)$ and $\phi(s, v) = \sum_{j \in \Delta(v)} z^v_{vj} + s_v$ denote the fractions of the objective function value $\Phi(a)$ (see (17d)) corresponding, respectively, to component $c \in \mathcal{C}(a)$ and to node $v \in \mathcal{V}(c)$. And let $S(s, c) = \sum_{v \in \mathcal{V}(c)} s_v$; note that $S(s, c) \leq V(c)$.

First, consider a single component $c \in \mathcal{C}(a)$. Notice that $\phi(s, v) = \min\{1 - s_v, S(s, c) - s_v\} + s_v$ for each $v \in \mathcal{V}(c)$. Thus, if $S(s, c) \leq 1$, then $\phi(s, v) = S(s, c) - s_v + s_v = S(s, c)$, and $\Phi(s, c) = V(c)S(s, c)$. And if $S(s, c) > 1$, then $\phi(s, v) = 1 - s_v + s_v = 1$ and $\Phi(s, c) = V(c)$. Therefore, one can define a new solution $s^* = (s^*_v)_{v \in \mathcal{V}}$ of (17d)-(17j) such that the values s^*_v of variables s_v are: 1 for arbitrary $\lfloor S(s, c) \rfloor$ nodes of c, $S(s, c) - \lfloor S(s, c) \rfloor$ for another arbitrary node of c, and 0 for the rest of nodes of c (note that the aggregated node flows defined by values of variables z^v_{ij} can always be redefined accordingly since c is a connected subgraph). Since $S(s^*, c) = S(s, c)$, that assignment yields $\Phi(s^*, c) = \Phi(s, c)$, and thus s^* is an optimal solution of (17d)-(17j) as well.

Let $s = (s_v)_{v \in \mathcal{V}}$ be an optimal solution derived by applying the above modification to all components $c \in \mathcal{C}(a)$. Notice that there cannot be two components $c', c'' \in \mathcal{C}(a)$ such that $S(s, c') > 1$ and $S(s, c'') < 1$, as one could define a new solution $s^* = (s^*_v)_{v \in \mathcal{V}}$ of (17d)-(17j) such that the values s^*_v of variables s_v would be: $s_v - \Delta s$ for node v' of c with a minimal strictly positive value of variable s_v, and $s_{v''} + \Delta s$ for a node v'' of c'' with the smallest (or zero) value of variable s_v (and values of variables z^v_{ij} would be redefined accordingly), where

$$\Delta s = \min\{s_{v'}, S(s, c') - 1, 1 - S(s, c'')\} > 0.$$

Since $S(s^*, c') + S(s^*, c'') = V(c') + S(s, c'') + \Delta s > S(s, c') + S(s, c'')$, that assignment would yield $\Phi(s^*, c) > \Phi(s, c)$, and thus s^* would be a better solution of (17d)-(17j) than s, which contradicts the optimality of s.

Next, consider two components $c', c'' \in \mathcal{C}(a)$ such that $S(s, c') - \lfloor S(s, c') \rfloor > 0$ and $S(s, c'') - \lfloor S(s, c'') \rfloor > 0$. W.l.o.g. assume that $V(c') \geq V(c'')$. Then, one can define a new solution $s^* = (s^*_v)_{v \in \mathcal{V}}$ of (17d)-(17j) such that the values s^*_v of variables s_v are: $S(s, c') - \lfloor S(s, c') \rfloor + \Delta s$ for the node of c' with a fractional value of s_v, and $S(s, c'') - \lfloor S(s, c'') \rfloor - \Delta s$ for the node of c'' with a fractional value of s_v (and values of variables z^v_{ij} are redefined accordingly), where

$$\Delta s = \min\{1 - (S(s, c') - \lfloor S(s, c') \rfloor), S(s, c'') - \lfloor S(s, c'') \rfloor\}.$$

Since $S^*(c') = S(s, c') + \Delta s$ and $S^*(c'') = S(s, c'') - \Delta s$, that assignment yields $\Phi(s^*, c') + \Phi(s^*, c'') \geq \Phi(s, c') + \Phi(s, c'')$, and thus s^* is also an optimal solution of (17d)-(17j), but the number of fractional values of $S(s, c)$ and of variables s_v is reduced at least by one.

Repeating the above procedure leads to an integer optimal solution of problem (17d)-(17j). Notice that there cannot be component $c \in \mathcal{C}(a)$ with a fractional value of $S(s, c)$, as one could define a new solution $s^* = (s^*_v)_{v \in \mathcal{V}}$ of (17d)-(17j) such that the value s^*_v of variable s_v would be 1 for the node of c with a fractional value of s_v (and values of variables z^v_{ij} would be redefined accordingly). Since $S(s^*, v) > S(s, v)$, that assignment would yield $\Phi(s^*, c) > \Phi(s, c)$, and thus s^* would be a better solution of (17d)-(17j) than s, which contradicts the optimality of s. ■

Theorem 3 Given a fixed attack $a = (a_v)_{v \in \mathcal{V}}$, the dual formulation (18) admits an optimal solution B, $t = (t_i^v)_{v,i \in \mathcal{V}}$ and $y = (y_i^v)_{v,i \in \mathcal{V}, i \neq v}$ where the vectors t and y are binary.

Proof: The proof of Theorem 3 is divided into two cases.

Case 1: Let us first consider the case where the maximum number of controllers M is less than or equal to the number of components after a, i.e., the case $M \le C(a)$. Then, as already observed in the first paragraph of Section 5.1, exactly M controllers are actually needed to obtain an optimal solution s of the considered follower's problem and each of them is placed in a separate (surviving) component (which means that $M = C(s, a)$).

Thus, let $s = (s_v)_{v \in \mathcal{V}}$ (where $\sum_{v \in \mathcal{V}} s_v = V(s) = M$) be an optimal (binary) solution of the MIP formulation (17d)-(17j) for a given fixed attack $a = (a_v)_{v \in \mathcal{V}}$. Clearly (see (3)), the sets $\mathcal{V}(a)$, $\mathcal{V}(c)$, $c \in C(s, a)$ and $\mathcal{V}(c)$, $c \in \overline{C}(s, a)$ form a partition of the set of nodes $\mathcal{V}$ into non-empty sets (recall that $C(s, a)$ is the family of the components surviving attack a and $\overline{C}(s, a)$ is the family of the remaining components induced by attack a).

Now, let $\mathcal{W} = \bigcup_{c \in C(s,a)} \mathcal{V}(c)$ denote the set of nodes surviving attack a when placement s is applied, and let $\overline{\mathcal{W}} = \bigcup_{c \in \overline{C}(s,a)} \mathcal{V}(c)$ be the set of nodes that although are not directly attacked do not survive. Clearly,

$$\mathcal{V}(a) \cup \mathcal{W} \cup \overline{\mathcal{W}} = \mathcal{V}.$$

Since, by assumption, each surviving component contains exactly one controller, we can define the sets $\mathcal{W}(w) = \mathcal{V}(c(w))$, $w \in \mathcal{V}(s)$, where $c(w) \in C(s, a)$ is the component containing controller w, and set the value of variable B to the size of the smallest surviving component(s):

$$B = \min \{ |\mathcal{W}(w)| : w \in \mathcal{V}(s) \}. \tag{23}$$

Further, for each $w \in \mathcal{V}(s)$ we select a set of nodes $\mathcal{T}(w) \subseteq \mathcal{W}(w)$ such that $T(w) = B$ for $w \in \mathcal{T}(w)$ (where $T(w) = |\mathcal{T}(w)|$) and the subgraph induced by $\mathcal{T}(w)$ in the graph of component $c(w)$ is connected, and define the set $\mathcal{T} = \bigcup_{w \in \mathcal{V}(s)} \mathcal{T}(w)$.

Now, let us define the first vector of dual variables $t = (t_i^v)_{(v,i) \in \mathcal{V}^2}$. For all $(v, i) \in \mathcal{V}^2$ such that $v = i$ we put

$$t_v^v = \begin{cases} 1, & \text{if } v \in \mathcal{V}(a) \cup (\mathcal{W} \setminus \mathcal{T}), \\ 0, & \text{if } v \in \mathcal{T} \cup \overline{\mathcal{W}}, \end{cases} \tag{24}$$

and for all $(v, i) \in \mathcal{V}^2$ such that $v \neq i$ we put

$$t_i^v = \begin{cases} 1, & \text{if } i \in \mathcal{V}(a) \text{ and } v \in \mathcal{V} \setminus \{i\} \\ 0, & \text{if } i \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } v \in \mathcal{V} \setminus \{i\}. \end{cases} \tag{25}$$

The second vector of dual variables $y = (y_i^v)_{(v,i) \in \mathcal{U}}$, where $\mathcal{U} = \mathcal{V}^2 \setminus \{(v, v) : v \in \mathcal{V}\}$, is defined as follows

$$y_i^v = \begin{cases} 1, & \text{if } i \in \mathcal{V}(a) \text{ and } v \in \mathcal{V} \setminus \{i\} \\ 1, & \text{if } i \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } v \in \mathcal{T} \cup \overline{\mathcal{W}} \text{ and } \exists c \in C(a) : \{i, v\} \subseteq \mathcal{V}(c) \\ 0, & \text{if } i \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } v \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } \forall c \in C(a) : \{i, v\} \nsubseteq \mathcal{V}(c) \\ 0, & \text{if } i \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } v \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } \exists w \in \mathcal{W} : i \in \mathcal{W}(w),\ v \in \mathcal{W}(w) \setminus \mathcal{T}(w) \\ 0, & \text{if } i \in \mathcal{W} \cup \overline{\mathcal{W}} \text{ and } v \in \mathcal{V}(a). \end{cases} \tag{26}$$

We claim that the variable B defined in this way and vectors of variables t and y form an optimal solution of the dual (18). The reasoning is as follows.

Let us first show that the so defined variables fulfil constraints (18b) for each node $i \in \mathcal{V}$. We consider four different cases covering all possible situations. The first case deals with a directly attacked node i, while the three other cases involve nodes that are not attacked. In the second and third cases, node i survives the attack and we distinguish whether it belongs to $\mathcal{T}$ (second case) or not (third case). Finally, the fourth case covers the situation where node i, although not directly attacked, does not survive.

- $i \in \mathcal{V}(a)$: By definitions (24)-(26), $t_i^i = t_i^v = y_i^v = 1$ for all $v \neq i$ so (18b) reduces to $B \geq 0$ which is obviously satisfied.
- $i \in \mathcal{T}(w)$ for some $w \in \mathcal{V}(s)$: $t_i^i = t_i^v = 0$ for all $v \neq i$, $y_i^v = 0$ for all $v \in \mathcal{V} \setminus \mathcal{T}(w)$, and $y_i^v = 1$ for all $v \in \mathcal{T}(w) \setminus \{i\}$. Thus,

$$B - \sum_{v \in \mathcal{V} \setminus \{i\}} y_i^v + \sum_{v \in V} t_i^v = B - \sum_{v \in \mathcal{T}(w) \setminus \{i\}} 1 = B - (T(w) - 1).$$

By definition (23), $B = T(w)$ and $B - (T(w) - 1) \geq 1$ so constraint (18b) is satisfied also in this case.

- $i \in \mathcal{W}(w) \setminus \mathcal{T}(w)$, for some $w \in \mathcal{V}(s)$: $t_i^i = 1$, $t_i^v = 0$ for all $v \neq i$, $y_i^v = 0$ for all $v \in \mathcal{V} \setminus \mathcal{T}(w)$, and $y_i^v = 1$ for all $v \in \mathcal{T}(w)\}$. Hence,

$$B - \sum_{v \in \mathcal{V} \setminus \{i\}} y_i^v + \sum_{v \in V} t_i^v = B - \sum_{v \in \mathcal{T}(w)} 1 + 1 = B - T(w) + 1 \geq 1.$$

- $i \in \overline{W}$: Now $t_i^i = 0$ and $t_i^v = 0$ for all $v \in \mathcal{V} \setminus i$. Let $c(i)$ denote the component in $\overline{C}(s, a)$ for which $i \in \mathcal{V}(c(i))$. Then, $y_i^v = 1$ for $v \in \mathcal{V}(c(i)) \setminus \{i\}$ and $y_i^v = 0$ for all nodes in $\mathcal{V} \setminus \mathcal{V}(c(i))$. Hence,

$$B - \sum_{v \in \mathcal{V} \setminus \{i\}} y_i^v + \sum_{v \in V} t_i^v = B - \sum_{v \in \mathcal{V}(c(i)) \setminus \{i\}} 1 = B - (V(c(i)) - 1) \geq 1.$$

The last inequality follows from the fact that for optimal solutions of the follower's problem the size of the smallest surviving component is not less than the size of the largest non-surviving component.

Let us now assume that there exists a link $\{v, j\} \in \mathcal{E}$ such that (18c) is not satisfied, i.e., $t_v^v = 0$ and $y_j^v = 0$. Then, $v \notin \mathcal{V}(a)$ by definition (24) and $j \notin \mathcal{V}(a)$ by definition (26), which means that link $\{v, j\}$ survives the attack. This implies that v and j belong to the same component and hence, again by (24) and (26), if $t_v^v = 0$ then $y_j^v = 1$ because then the condition on v and j in the second case of (26) is fulfilled. This is a contradiction, so vectors t and y must satisfy constraint (18c).

To show that constraints (18d) and (18e) are satisfied by the above defined vectors t and y, consider an arbitrary link $\{i, j\} \in \mathcal{E}$ and an arbitrary node $v \notin \{i, j\}$. If $i \in \mathcal{V}(a)$, then $t_i^v = y_i^v = 1$ and (18d) and (18e) are trivially satisfied. A symmetric argument holds if $j \in \mathcal{V}(a)$. The only case remaining is when $i \notin \mathcal{V}(a)$ and $j \notin \mathcal{V}(a)$, which implies that $t_i^v = t_j^v = 0$ and (18d) and (18e) reduce to $y_i^v = y_j^v$. Clearly, if $v \in \mathcal{V}(a)$, then $y_i^v = y_j^v = 1$. If $v \notin \mathcal{V}(a)$, assume $y_i^v = 1$. Then i and v belong to the same connected component. As link $\{i, j\}$ survives the attack, j also belongs to this component, hence $y_j^v = 1$. By symmetry, $y_j^v = 1$ implies $y_i^v = 1$ so we can finally conclude that (18d) and (18e) are satisfied.

We finish the proof of Case 1 by showing that the objective value of the constructed feasible dual solution B, t, y is optimal. Indeed,

$$\begin{aligned}
&MB + \sum_{i\in\mathcal{V}}\sum_{v\in\mathcal{V}}(1-a_i)t_i^v \\
&= \sum_{w\in\mathcal{V}(s)} |\mathcal{T}(w)| + \sum_{i\in\mathcal{V}}(1-a_i)t_i^i + \sum_{i\in\mathcal{V}}\sum_{v\in\mathcal{V}\setminus\{i\}}(1-a_i)a_i \\
&= |\mathcal{T}| + \sum_{i\in\mathcal{V}\setminus\mathcal{V}(a)} t_i^i \;\; = |\mathcal{T}| + \sum_{i\in\mathcal{W}\setminus\mathcal{T}} t_i^i \;\; = |\mathcal{T}| + |W \setminus \mathcal{T}| \\
&= |\mathcal{T}| + |\mathcal{W}| - |\mathcal{T}| \;\; = |\mathcal{W}|,
\end{aligned}$$

hence the objective function of the dual solution is equal to the number of surviving nodes.

Case 2: Now, let us consider the case when $M \geq C(a)$, i.e., when there are more controllers than the number of components surviving the attack. Then, as also mentioned in the first paragraph of Section 5.1, it is sufficient to use $C(a)$ controllers, one in each component of $\mathcal{C}(a)$, and consider an optimal placement s with $V(s) = C(a)$. Clearly, the number of surviving nodes for any such placement is equal to its upper bound, i.e., to $V - K$.

Now let us set $B = 0$ and define the values of the remaining dual variables as follows.

$$t_i^i = 1, \qquad i \in \mathcal{V} \tag{27a}$$

$$t_i^v = y_i^v = a_i, \qquad i \in \mathcal{V},\ v \in \mathcal{V} \setminus \{i\}. \tag{27b}$$

It is easy to check that the so defined variables form an optimal solution of the dual (18). Since $B = 0$, constraint (18b) takes the form

$$t_i^i + \sum_{v\in V\setminus\{i\}} t_i^v \geq 1 + \sum_{v\in\mathcal{V}\setminus\{i\}} y_i^v, \qquad i \in \mathcal{V},$$

which is obviously satisfied by the values assigned to t and y in (27). Then, constraint (18c) is satisfied due to (27a), and constraints (18d) and (18e) reduce, due to (27b), to $y_j^v \geq 0$ and $y_i^v \geq 0$.

Finally, the value of the dual objective function for the so defined dual variables, is as follows

$$\begin{aligned}
&MB + \sum_{i\in\mathcal{V}}\sum_{v\in\mathcal{V}}(1-a_i)t_i^v \\
&= \sum_{i\in\mathcal{V}}(1-a_i)t_i^i + \sum_{i\in\mathcal{V}}\sum_{v\in\mathcal{V}\setminus\{i\}}(1-a_i)a_i \\
&= V - K,
\end{aligned}$$

as required.

Finally, let us note that in the case $M = C(a)$ both the solution of Case 1 (with $B > 0$) and the solution of Case 2 (with $B = 0$) are valid. Actually, the complementary slackness condition applied to constraint (17e) of the linear relaxation of the follower's problem implies that in any optimal solution of the dual the value of B must be equal to 0 when $M > C(a)$), and can be greater or equal to 0 when $M \leq C(a)$. ■

B | NODE NUMBERING IN THE *COST266* AND *CORONET CONUS* NETWORKS

TABLE 13 *cost266*: node numbers

1	Berlin	14	Stockholm	27	Barcelona
2	Palermo	15	Lisbon	28	Lyon
3	Seville	16	Madrid	29	London
4	Cracow	17	Helsinki	30	Dusseldorf
5	Strasbourg	18	Copenhagen	31	Zagreb
6	Paris	19	Zurich	32	Sofia
7	Warsaw	20	Oslo	33	Amsterdam
8	Belgrade	21	Marseille	34	Dublin
9	Milan	22	Frankfurt	35	Brussels
10	Vienna	23	Rome	36	Munich
11	Birmingham	24	Hamburg	37	Glasgow
12	Budapest	25	Athens		
13	Prague	26	Bordeaux		

TABLE 14 *coronet conus*: node numbers

1	Abilene	26	Houston	51	Providence
2	Albany	27	Jacksonville	52	Raleigh
3	Albuquerque	28	Kansas City	53	Richmond
4	Atlanta	29	Las Vegas	54	Rochester
5	Austin	30	Little Rock	55	Sacramento
6	Baltimore	31	Long Island	56	Salt Lake City
7	Baton Rouge	32	Los Angeles	57	San Antonio
8	Billings	33	Louisville	58	San Diego
9	Birmingham	34	Memphis	59	San Francisco
10	Bismarck	35	Miami	60	San Jose
11	Boston	36	Milwaukee	61	Santa Barbara
12	Buffalo	37	Minneapolis	62	Scranton
13	Charleston	38	Nashville	63	Seattle
14	Charlotte	39	New Orleans	64	Spokane
15	Chicago	40	New York	65	Springfield
16	Cincinnati	41	Newark	66	St Louis
17	Cleveland	42	Norfolk	67	Syracuse
18	Columbus	43	Oakland	68	Tallahassee
19	Dallas	44	Oklahoma City	69	Tampa
20	Denver	45	Omaha	70	Toledo
21	Detroit	46	Orlando	71	Tucson
22	El Paso	47	Philadelphia	72	Tulsa
23	Fresno	48	Phoenix	73	Washington DC
24	Greensboro	49	Pittsburgh	74	West Palm Beach
25	Hartford	50	Portland	75	Wilmington

C | COMPUTATION TIMES

Below are other tables (in addition to those shown in Sections 7.2 and 7.3) showing the computation times related the results discusses in Section 7.

C.1 | Computations time for *cost266*

The following tables show the computation times consumed by $\mathbb{F}1$, $\mathbb{F}3$,$\mathbb{F}4$ and $\mathbb{F}1R$, $\mathbb{F}3R$,$\mathbb{F}4R$ when optimizing the *cost266* network (the computation times for $\mathbb{F}2$ and $\mathbb{F}2R$ are shown in Section 7.2).

TABLE 15 *cost266*: computation times for $\mathbb{F}3$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.12	0.10	0.08	0.07	0.10	0.08	0.07	0.09	0.07	0.07
$K = 2$	0.23	0.20	0.20	0.23	0.24	0.35	0.21	0.32	0.19	0.16
$K = 3$	0.46	0.56	0.58	0.61	0.53	0.53	0.74	0.65	0.48	0.48
$K = 4$	0.97	0.84	1.96	1.37	1.77	2.37	1.11	0.99	1.12	1.09
$K = 5$	2.36	3.28	4.14	4.15	6.14	6.62	4.82	3.96	3.67	3.61
$K = 6$	5.19	5.99	8.59	9.60	10.7	10.6	11.5	6.53	6.36	6.55
$K = 7$	6.95	7.98	12.4	10.2	15.5	77.7	73.3	13.2	11.5	10.0
$K = 8$	6.24	10.1	16.3	16.4	73.5	70.2	84.8	96.0	20.4	18.5
$K = 9$	11.5	16.7	17.6	20.9	74.9	85.1	60.5	59.9	108	61.5
$K = 10$	54.1	19.2	20.6	22.8	23.4	63.9	90.1	50.7	108	59.9

TABLE 16 *cost266*: computation times for $\mathbb{F}4$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	2.46	1.65	1.19	1.01	0.98	1.04	0.75	0.87	0.74	0.87
$K = 2$	1.79	3.41	3.13	2.38	1.94	1.78	1.50	1.41	0.78	0.72
$K = 3$	0.59	4.45	7.72	7.28	5.76	4.18	3.06	3.91	2.95	2.77
$K = 4$	1.94	4.01	8.61	8.21	16.4	12.2	12.3	8.26	5.96	5.14
$K = 5$	1.19	4.97	7.64	27.7	31.2	21.0	16.2	12.6	10.3	8.64
$K = 6$	3.36	3.48	6.94	26.1	25.2	45.6	30.1	20.7	14.7	11.0
$K = 7$	2.04	1.60	3.77	25.1	20.9	31.5	62.7	37.3	26.4	16.5
$K = 8$	2.72	2.71	4.08	7.30	20.1	21.0	36.6	84.1	51.7	32.2
$K = 9$	3.55	3.02	1.69	4.33	7.48	20.4	24.2	39.5	91.0	47.3
$K = 10$	4.36	2.70	3.45	3.55	5.23	10.4	21.9	23.5	36.6	90.9

TABLE 17 *cost266*: computation times for $\mathbb{F}3R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.07	0.06	0.07	0.06	0.06	0.06	0.06	0.07	0.06	0.06
$K = 2$	0.12	0.11	0.12	0.15	0.19	0.09	0.07	0.08	0.07	0.06
$K = 3$	0.15	0.28	0.27	0.36	0.34	0.34	0.27	0.16	0.16	0.19
$K = 4$	0.38	0.43	0.53	0.50	0.49	0.35	0.35	0.34	0.52	0.29
$K = 5$	1.04	1.18	1.20	2.10	1.58	1.71	1.39	1.34	1.40	1.45
$K = 6$	2.44	1.69	3.18	3.53	4.57	4.51	3.73	2.64	3.00	2.26
$K = 7$	2.90	2.95	2.35	5.86	5.18	7.41	13.76	6.24	5.16	3.46
$K = 8$	2.66	3.57	3.83	5.47	11.2	12.4	15.8	15.4	10.1	7.89
$K = 9$	3.69	3.83	4.71	5.11	13.1	13.5	14.3	12.1	14.1	13.9
$K = 10$	3.66	4.59	6.46	5.86	10.1	10.8	12.9	10.6	9.10	12.1

TABLE 18 *cost266*: computation times for $\mathbb{F}4R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	1.74	1.94	0.88	0.76	0.77	0.74	0.57	0.38	0.36	0.30
$K = 2$	1.12	1.87	1.83	0.96	0.64	0.69	0.55	0.57	0.51	0.38
$K = 3$	0.51	2.86	3.09	2.76	1.99	1.64	1.72	0.92	0.72	0.71
$K = 4$	1.43	2.23	3.78	3.30	4.38	4.17	2.62	1.80	1.52	1.20
$K = 5$	1.24	2.85	3.89	8.45	15.90	10.16	7.82	3.96	2.92	1.98
$K = 6$	2.35	1.32	3.44	7.64	13.92	15.24	9.39	8.38	5.95	3.06
$K = 7$	0.75	1.06	2.21	7.59	7.85	13.7	17.4	11.6	8.03	5.11
$K = 8$	1.08	1.01	0.97	2.82	7.40	10.3	12.3	19.8	9.97	6.20
$K = 9$	1.52	0.89	0.89	1.09	1.95	6.27	9.23	11.8	17.4	9.05
$K = 10$	2.11	1.38	1.32	1.03	0.89	2.35	9.24	7.86	10.4	14.1

TABLE 19 *cost266*: computation times for $\mathbb{F}1$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.04	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00
$K = 2$	0.13	0.12	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.01
$K = 3$	0.12	0.14	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
$K = 4$	1.06	1.10	0.91	0.81	0.01	0.01	0.02	0.01	0.01	0.01
$K = 5$	1.20	1.86	1.48	1.63	0.02	0.01	0.01	0.02	0.02	0.01
$K = 6$	10.7	14.7	18.0	14.2	14.3	0.02	0.02	0.01	0.02	0.01
$K = 7$	34.2	42.2	54.5	73.8	46.3	37.2	0.02	0.02	0.02	0.02
$K = 8$	223	227	230	300	300	300	300	0.02	0.02	0.02
$K = 9$	300	300	300	300	300	300	300	300	0.02	0.02
$K = 10$	300	300	300	300	300	300	300	300	300	0.03

TABLE 20 *cost266*: computation times for $\mathbb{F}1R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
$K = 2$	0.10	0.14	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02
$K = 3$	0.13	0.13	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.02
$K = 4$	0.84	0.96	1.15	0.98	0.02	0.01	0.02	0.01	0.03	0.01
$K = 5$	1.27	1.45	1.51	1.80	0.01	0.02	0.01	0.02	0.02	0.01
$K = 6$	10.5	14.9	13.6	15.0	11.8	0.02	0.02	0.02	0.02	0.02
$K = 7$	24.7	25.3	50.9	57.7	41.2	49.0	0.04	0.02	0.02	0.03
$K = 8$	125	106	158	240	220	167	177	0.02	0.02	0.02
$K = 9$	300	300	300	300	300	300	300	300	0.03	0.04
$K = 10$	300	300	300	300	300	300	300	300	300	0.03

C.2 | Computation times for *coroner conus*

The following tables show the computation times consumed by $\mathbb{F}1$, $\mathbb{F}3$,$\mathbb{F}4$ and $\mathbb{F}1R$, $\mathbb{F}3R$,$\mathbb{F}4R$ when optimizing the *coronet conus* network (the computation times for $\mathbb{F}2$ and $\mathbb{F}2R$ are shown in Section 7.2).

TABLE 21 *coronet conus*: computation times for $\mathbb{F}3$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.36	0.33	0.33	0.32	0.37	0.35	0.35	0.33	0.34	0.34
$K = 2$	2.13	1.96	1.93	2.12	1.76	2.31	1.86	1.99	1.84	2.28
$K = 3$	14.7	16.7	21.9	23.4	26.8	19.8	21.7	19.6	22.2	25.7
$K = 4$	33.9	49.7	68.2	326	81.9	75.5	70.6	67.3	73.0	69.5
$K = 5$	52.7	86.1	113	600	600	600	246	190	196	176
$K = 6$	85.8	83.7	600	600	600	600	600	600	600	600
$K = 7$	215	245	600	600	600	600	600	600	600	600
$K = 8$	600	600	600	600	600	600	600	600	600	600

TABLE 22 *coronet conus*: computation times for $\mathbb{F}4$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	29.8	38.8	29.4	19.1	18.0	9.34	13.0	12.2	8.93	9.93
$K = 2$	52.3	209	114	81.2	63.6	66.5	61.0	66.9	57.0	54.7
$K = 3$	135	600	600	600	600	517	448	356	304	277
$K = 4$	37.0	600	600	600	600	600	600	600	600	600
$K = 5$	86.3	539	600	600	600	600	600	600	600	600
$K = 6$	47.6	93.5	600	600	600	600	600	600	600	600
$K = 7$	38.9	69.1	600	600	600	600	600	600	600	600
$K = 8$	45.8	59.0	600	600	600	600	600	600	600	600

TABLE 23 *coronet conus*: computation times for $\mathbb{F}3R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	0.40	0.34	0.34	0.33	0.33	0.34	0.34	0.35	0.34	0.34
$K = 2$	0.57	0.54	0.69	0.70	0.64	0.63	0.49	0.61	0.55	0.62
$K = 3$	2.56	2.40	2.66	3.83	3.08	3.24	4.34	3.47	3.89	3.59
$K = 4$	8.37	11.4	13.4	20.0	19.4	21.4	17.8	21.4	19.6	16.4
$K = 5$	19.6	37.3	41.5	58.7	159	79.2	75.1	85.4	75.5	74.9
$K = 6$	40.1	48.7	158	220	252	315	237	348	194	242
$K = 7$	81.3	159	219	267	319	458	600	600	506	507
$K = 8$	299	181	261	251	492	600	600	600	600	600

TABLE 24 *coronet conus*: computation times for $\mathbb{F}4R$

time [s]	$M = 1$	$M = 2$	$M = 3$	$M = 4$	$M = 5$	$M = 6$	$M = 7$	$M = 8$	$M = 9$	$M = 10$
$K = 1$	21.4	19.3	11.9	8.84	6.95	4.62	5.74	5.53	5.20	3.92
$K = 2$	30.9	50.1	28.5	24.0	17.8	17.2	12.4	10.4	8.19	8.46
$K = 3$	52.9	80.3	140.2	101	69.6	49.7	40.6	32.7	27.6	22.0
$K = 4$	10.9	101	322	600	358	259	190	142	104	81.3
$K = 5$	29.5	66.5	140	368	600	600	600	450	333	244
$K = 6$	18.1	20.7	361	436	550	600	600	600	600	600
$K = 7$	9.92	25.8	169.4	439	600	600	600	600	600	600
$K = 8$	8.81	14.9	89.2	111	457	600	600	600	600	600