Max-Min Optimization of Controller Placements vs. Min-Max Optimization of Attacks on Nodes in Service Networks

Artur Tomaszewski
Warsaw University of Technology
Warsaw, Poland
a.tomaszewski@tele.pw.edu.pl

Michał Pióro
Warsaw University of Technology
Warsaw, Poland
m.pioro@tele.pw.edu.pl

Mariusz Mycek
Warsaw University of Technology
Warsaw, Poland
m.mycek@tele.pw.edu.pl

ABSTRACT

The paper deals with two complementary optimization problems related to the resilience of communication networks against targeted node attacks, where the proper functioning of the network requires that the nodes are connected to the so called controllers that are placed in selected node locations – a node that looses such a connection in the result of an attack is considered lost. These two problems can be used to optimize (maximize in the case of a network operator and minimize in the case of an attacker) the resilience in question when the interaction between these two parties is considered within the framework of game theory. The presented formulations and their solution algorithms are original. The efficiency of the algorithms is illustrated for a medium size network by means of a numerical example.

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 controller 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 node using some network path. That is why the locations are interconnected with transport links.

Such a setting is directly applicable, in particular, to software defined networks (SDN) [2, 3]. Note however that it may also arise in a number of other contexts. In the ICT area it may also apply to content delivery networks (CDN): delivery nodes, which deliver content to the user, correspond to the service nodes, and origin nodes, which are the primary sources of the original content, correspond to the control nodes (storage node, which are responsible for storing copies of original data, may correspond either to the service nodes or to the control nodes). One may find applications of the considered model in other areas as well, would it be utilities, manufacturing, logistics, sales, or medicine. In those areas 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 utility and transportation networks, and their inaccessibility may result not only from attacks, but, e.g., from technical malfunctioning and natural disasters.

We aim at protecting the network against targeted node attacks. The attack targets a set of selected locations making both service nodes and controller nodes (if any) at those locations unavailable. Moreover, the attack makes unavailable the transport 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 will still provide service (and will be called a surviving (service) node) only if its location has not been attacked and the component it belongs to still contains at least one location with a controller node.

In the paper we consider two complementary optimization problems for the so described network layout. The first of them consists in finding a placement of a given number of controllers that maximizes the number of nodes that survive the worst case attack from a given, in general non-compact, list of attacks. The complementary problem, in turn, is to find an attack targeted at a given number of locations that minimizes the number of surviving nodes for any placement from a given list (also in general non-compact) of controller placements. We note here that although related problems have been widely researched in the literature (mainly in the context of SDN, see [1, 5–7] and references therein), the two problem formulations and algorithms for solving them presented below are original.

2 NOTATION AND PROBLEM DESCRIPTION

2.1 Notation

First, we model the service network by means of a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V} = \{1, 2, \ldots, V\}$ represents network locations, and $\mathcal{E}$ ($\mathcal{E} \subseteq \{\mathcal{X} \subseteq \mathcal{V} : |\mathcal{X}| = 2\}$) is the set of transport links represented by unordered nodes pairs that interconnect the locations; for each $e \in \mathcal{E}$, let $\alpha(e), \beta(e) \in \mathcal{V}$ denote the end nodes of link e, and for each $v \in \mathcal{V}$, let $\delta(v) = \{e \in \mathcal{E} : v \in \{\alpha(e), \beta(e)\}\}$ denote the set of links incident with node v.

Next, we assume that the network is equipped with controllers and the set of (allowable) controller placements is denoted by $\mathcal{S}$. Each placement $s \in \mathcal{S}$ is characterized by the set $\mathcal{V}(s)$ ($\mathcal{V}(s) \subseteq \mathcal{V}$) where the controllers are actually placed (apart form the service nodes). A typical example of the set of placements $\mathcal{S}$ is the set of all M-node placements (where $0 < M \leq V$), i.e., the set of all placements s with $|\mathcal{V}(s)| = M$; such a set will be denoted by $\mathcal{S}(M)$.

Then, we consider a set $\mathcal{A}$ of attacks targeted at networks nodes. Each attack $a \in \mathcal{A}$, is characterized by the set of the attacked locations $\mathcal{V}(a)$ ($\mathcal{V}(a) \subseteq \mathcal{V}$), which defines the set $\mathcal{C}(a)$ of (non-empty) connected components into which the network graph $\mathcal{G}$

Table 1: Summary of notation.

$\mathcal{V}, \mathcal{E}$	sets of nodes and links ($V = \lvert\mathcal{V}\rvert, E = \lvert\mathcal{E}\rvert$)
$\alpha(e), \beta(e)$	end nodes of link $e \in \mathcal{E}$
$\delta(v)$	set of links incident with node $v \in \mathcal{V}$
$\mathcal{S}$	set of allowable controller placements
$\mathcal{V}(s)$	set of controller nodes locations in placement $s \in \mathcal{S}$
$\mathcal{A}$	set of expected attacks
$\mathcal{V}(a)$	set of nodes affected by attack $a \in \mathcal{A}$
$\mathcal{C}(a)$	set (family) of components induced by attack $a \in \mathcal{A}$
$\mathcal{V}(c)$	set of nodes of component $c \in \mathcal{C}(a)$
$\mathcal{S}(M)$	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 $a \in \mathcal{A}$
$\mathbb{R}^+$	set of nonnegative real numbers

is split as the result of attack a. For each component $c \in \mathcal{C}(a)$, $\mathcal{V}(c)$ will denote the set of its nodes. A typical example of the set of attacks $\mathcal{A}$ is the set of all K-node attacks, i.e., the set of all attacks a with $|\mathcal{V}(a)| = K$, for a given integer parameter K (where $0 < K < V$); such a set will be denoted by $\mathcal{A}(K)$.

Finally, we assume that as a result of an attack a in each (directly attacked) location in $\mathcal{V}(a)$ its service node and the 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 components in $\mathcal{C}(a)$ that contain a controller.

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. For such a measure, we can introduce the notions of the worst attack and the best controller 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 controller placement s in $\mathcal{S}$ that maximizes the value of $V(s, a)$, i.e., $V(s(a), a) = \max_{s\in\mathcal{S}} V(s, a)$. (Certainly, there can be multiple worst attacks and multiple best placements.)

2.2 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 approaches for finding controller placements (the operator) and for constructing attacks (the attacker). In this paper we introduce a mathematical model aimed at solving optimization problems related to these issues.

The optimization problems we consider stem from the assumption that the set of possible controller placements $\mathcal{S}$ and the set of possible attacks $\mathcal{A}$ are known to both the operator and the attacker, and each of them is trying to find a solution that is most effective in the case of the worst attacks (the operator) and the best placements (the attacker). Hence, it is natural to consider 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 $\mathcal{S}$:

$$V(s^*, a(s^*)) = \max_{s\in\mathcal{S}} V(s, a(s)) = \max_{s\in\mathcal{S}} \min_{a\in\mathcal{A}} V(s, a). \quad (1)$$

Each placement s^* that solves problem (1) will be 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 equal at least to Y^* for any attack in $\mathcal{A}$, where Y^* 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 $\mathcal{A}$:

$$V(s(a^*), a^*) = \min_{a\in\mathcal{A}} V(s(a), a) = \min_{a\in\mathcal{A}} \max_{s\in\mathcal{S}} V(s, a). \quad (2)$$

Each attack a^* that solves problem (2) will be called the worst attack for a given set of placements $\mathcal{S}$. Thus, attack a^* guarantees that the number of surviving nodes is equal at most to Z^* for any placement in $\mathcal{S}$, where Z^* is the minimum number with this property.

3 OPTIMIZATION PROBLEMS

In this section we will present integer programming (IP) formulations of the two basic optimization problems (CPOP and NAOP) described in Section 2.2, with an intention to be able to solve them using commercial IP solvers.

3.1 Max-min controller placement optimization problem (CPOP)

In the formulation of CPOP presented below we assume that $\mathcal{S} = \mathcal{S}(M)$ (i.e., we consider the set of all M-node placements) and $\mathcal{A}$ is an arbitrary set of attacks. This means that we consider the problem of finding an M-node controller placement that maximizes the number of nodes surviving its worst attack from set $\mathcal{A}$.

Let s_v ($v \in \mathcal{V}$) be a binary variable that equals 1 if, and only if, a network controller is placed at location v. (Each vector $s = (s_v)_{v\in\mathcal{V}}$ specifies placement s with $\mathcal{V}(s) = \{v \in \mathcal{V} : s_v = 1\}$.) And for all $a \in \mathcal{A}$, $v \in \mathcal{V}$, let y_v^a be a binary variable that equals 1 if, and only if, service node at location v survives attack a. The formulation (abbreviated by $\mathbb{P}[M, \mathcal{A}]$) is as follows:

$$\mathbb{P}[M, \mathcal{A}] : \max\ Y \quad (3a)$$

$$\sum_{v\in\mathcal{V}} s_v = M \quad (3b)$$

$$y_v^a = 0 \qquad a \in \mathcal{A},\ v \in \mathcal{V}(a) \quad (3c)$$

$$\sum_{v\in\mathcal{V}(c)} y_v^a \le |\mathcal{V}(c)| \sum_{v\in\mathcal{V}(c)} s_v \qquad a \in \mathcal{A},\ c \in \mathcal{C}(a) \quad (3d)$$

$$Y \le \sum_{v\in\mathcal{V}} y_v^a \qquad a \in \mathcal{A} \quad (3e)$$

$$s_v, y_v^a \in \{0, 1\} \qquad a \in \mathcal{A},\ v \in \mathcal{V} \quad (3f)$$

$$Y \in \mathbb{R}^+. \quad (3g)$$

Constraint (3b) sets the number of deployed controllers to M. Then, constraints (3c) explicitly force variables y_v^a with node v directly destroyed by attack a to be equal to 0 (these nodes do not survive after attack a wherever the controllers are placed).

Constraints (3d), in turn, imply that when a component c induced by attack a does not contain any controller ($\sum_{v\in\mathcal{V}(c)} s_v = 0$) then

all nodes in c do not survive and hence the corresponding variables y_v^a are explicitly set to 0 (because $\sum_{v\in\mathcal{V}(c)} y_v^a$ is forced to be equal to 0). Otherwise, when c contains at least one controller, then the right-hand side of (3d) is greater than or equal to the number of elements in component c and hence it allows all y_v^a with v in c to be greater than 0 (but not greater than 1 since these are binary variables). Now we note that for any fixed vector of controller placement variables $s = (s_v)_{v\in\mathcal{V}}$, optimization objective (3a) and constraints (3e) will force the value of variable Y to be equal to the actual number of surviving nodes after at least one attack a for which this number is minimal over $\mathcal{A}$, because for such an attack the values of those variables y_v^a that are not explicitly set to 0 will reach their maximum, i.e., 1.

Hence, when variables s are optimized, the final value Y^* of the objective function will be equal to the maximum, over all placements in $\mathcal{S}(M)$, of the number of surviving nodes (i.e., the total number of nodes appearing in the components containing one or more controllers) when the worst attack is assumed for each of the considered placements.

In summary, the maximum value Y^* of objective (3a) is equal to

$$V(s^*, a(s^*)) = \max_{s\in\mathcal{S}(M)} \min_{a\in\mathcal{A}} V(s,a), \tag{4}$$

where s^* denotes an arbitrary optimal placement resulting from (3); this means that any optimal s^* is one of the best placements in $\mathcal{S}(M)$ with respect to the set of attacks $\mathcal{A}$.

3.2 Min-max node attack optimization problem (NAOP)

In the formulation of NAOP presented below we assume that $\mathcal{A} = \mathcal{A}(K)$ (i.e., we consider the set of all K-node attacks) and $\mathcal{S}$ is an arbitrary set of 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.

In the formulation (abbreviated by $\mathbb{A}[K,\mathcal{S}]$), 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 $a = (a_v)_{v\in\mathcal{V}}$ specifies attack a with $\mathcal{V}(a) = \{v\in\mathcal{V} : a_v = 1\}$.) Next, for each $e\in\mathcal{E}$, t_e is a binary variable that equals 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_v^s is a binary variable equal to 1 if, and only if, node v survives the constructed attack when controller placement s is assumed. The formulation uses the fact that if after the attack a node can still provide service, 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, and is as follows:

$$\mathbb{A}[K,\mathcal{S}]: \min\ Z \tag{5a}$$
$$\sum_{v\in\mathcal{V}} a_v = K \tag{5b}$$
$$t_e \geq a_v \qquad v\in\mathcal{V}, e\in\delta(v) \tag{5c}$$
$$t_e \leq a_{\alpha(e)} + a_{\beta(e)} \qquad e\in\mathcal{E} \tag{5d}$$
$$z_v^s \leq 1 - a_v \qquad s\in\mathcal{S}, v\in\mathcal{V} \tag{5e}$$
$$z_v^s \geq 1 - a_v \qquad s\in\mathcal{S}, v\in\mathcal{V}(s) \tag{5f}$$
$$z_{\alpha(e)}^s \geq z_{\beta(e)}^s - t_e \qquad s\in\mathcal{S}, e\in\mathcal{E} \tag{5g}$$
$$z_{\beta(e)}^s \geq z_{\alpha(e)}^s - t_e \qquad s\in\mathcal{S}, e\in\mathcal{E} \tag{5h}$$
$$Z \geq \sum_{v\in\mathcal{V}} z_v^s \qquad s\in\mathcal{S} \tag{5i}$$
$$a_v, t_e, z_v^s \in \{0,1\} \qquad s\in\mathcal{S}, v\in\mathcal{V}, e\in\mathcal{E} \tag{5j}$$
$$Z\in\mathbb{R}^+. \tag{5k}$$

Constraint (5b) sets the number of attacked nodes to K. Then, constraints (5c) and (5d) force, as required, the value of each binary variable t_e to be equal to 0 (which means that link e is available after attack a) if, and only if, both end nodes of e are not directly attacked. Next, constraints (5e) set z_v^s to 0 (which means that node v does not survive attack a if node v is attacked directly, whatever placement is selected. Constraints (5f), in turn, set z_v^s to 1 (which means that node v survives attack a when placement s is assumed) if node v is not directly attacked and its location contains a controller.

The next two sets of constraints, (5g) and (5h), make sure 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. This property assures that all nodes in any component $c\in C(a)$ (i.e., in any component c resulting from the constructed attack a) have the same values of z_v^s (for any fixed placement s):

$$z_v^s = z_w^s, \quad v,w\in c, c\in C(a), s\in\mathcal{S}. \tag{6}$$

Moreover, for any given placement $s\in\mathcal{S}$, if a location $v\in\mathcal{V}(c)$ contains a controller then the inequality in (5f) sets z_v^s to 1 since, by definition, this location is not attacked. In this case the equalities in (6) 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 (5i).

On the other hand, when component $c\in C(a)$ 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, however, is not an issue. To see this consider an optimal solution of (5) and let $\mathcal{S}^*$ denote the set of all placements in $\mathcal{S}$ for which the inequalities in (5i) 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

$$Z^* = \sum_{v\in\mathcal{V}} z_v^s = \sum_{c\in C(a^*)} \sum_{v\in\mathcal{V}(c)} z_v^s, \quad s\in\mathcal{S}^* \tag{7}$$

(because $\mathcal{V} = \mathcal{V}(a^*) \cup \bigcup_{c\in C(a^*)} \mathcal{V}(c)$ 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 $c\in C(a^*)$ 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 argumentation reveals that in fact constraints (5e) are redundant.)

In summary, the minimum value Z^* of objective (5a) is equal to

$$V(s(a^*), a^*) = \min_{a\in\mathcal{A}(K)} \max_{s\in\mathcal{S}} V(s,a), \tag{8}$$

where a^* denotes an arbitrary optimal attack resulting from (5), that is one of the worst attacks in $\mathcal{A}(K)$ for the assumed set of placements $\mathcal{S}$.

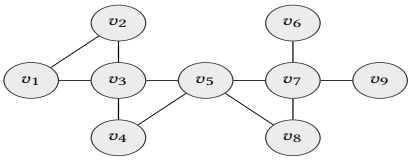

Figure 1: Sample 9-node and 11-link network.

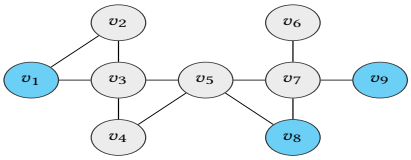

Figure 2: One of 32 best placements (with $\mathcal{V}(s) = \{1, 8, 9\}$).

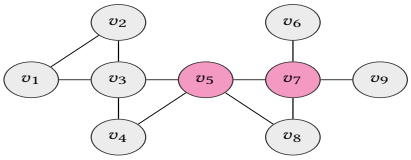

Figure 3: One of 2 worst attacks (with $\mathcal{V}(a) = \{5, 7\}$).

3.3 An example

We will now characterize solutions of formulations $\mathbb{P}[M, \mathcal{A}(K)]$ and $\mathbb{A}[K, \mathcal{S}(M)]$ for the network depicted in Figure 1 and the 3-node placements ($M = 3$) and the 2-node attacks ($K = 2$).

Then, 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 will survive whatever 3-node 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 will survive whatever 2-node attack is selected. We do not list all 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 will select 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 belonging to the set of 32 best placements with respect to $\mathcal{A}(2)$), and each of them guarantees that at least 6 nodes will survive any of the two considered attacks. Note that this value is substantially better 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 will survive if any of the placements in the set $\mathcal{S} = \{s^1, s^2, s^3, s^4\}$ is deployed.

4 SOLVING NON-COMPACT VERSIONS OF CPOP AND NAOP

Note that both formulations $\mathbb{P}[M, \mathcal{A}]$ and $\mathbb{A}[K, \mathcal{S}]$ are in general non-compact as the number of attacks in the set $\mathcal{A}$ appearing in the first formulation, and the number of placements in the set $\mathcal{S}$ appearing in the second formulation may grow exponentially with the number of nodes V. When this is the case, CPOP can be approached using an attack generation procedure (provided $\mathcal{A}$ can be characterized in a tractable way) while NAOP can be approached using a controller placement generation procedure (provided $\mathcal{S}$ can be characterized in a tractable way).

Such non-compactness can appear for $\mathcal{A} = \mathcal{A}(K)$ (for example when $V = 2K$), and for $\mathcal{S} = \mathcal{S}(M)$ (for example when $V = 2M$). Therefore, below we present an algorithm for solving formulation $\mathbb{P}[M, \mathcal{A}(K)]$ (Section 4.1) and a similar algorithm for solving formulation $\mathbb{A}[K, \mathcal{S}(M)]$ (Section 4.2).

4.1 Solving $\mathbb{P}[M, \mathcal{A}(K)]$ by attack generation

In the algorithm, the list of attacks $\mathcal{A}$ is iteratively extended by means of solving consecutive formulations of NAOP of the form $\mathbb{A}[K, \{s^*\}]$ for a particular placement s^* (NAOP is called the *pricing problem* in this context), and at each iteration, for the current list $\mathcal{A}$, an optimal placement s^* is found by means of solving formulation $\mathbb{P}[M, \mathcal{A}]$ (CPOP is called the *master problem* in this context). In effect, in each iteration we find out whether there exists an attack $a \in \mathcal{A}(K) \setminus \mathcal{A}$ such that when a is added to the current list of attacks $\mathcal{A}$, then the number of surviving nodes after attack a is smaller than the maximum number of surviving nodes guaranteed for any attack in $\mathcal{A}$ (achieved for the current optimal placement). If this is the case, we re-optimize s^* and continue.

A1: Algorithm for controller placement optimization by means of attack generation

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

4.2 Solving $\mathbb{A}[K, \mathcal{S}(M)]$ by controller placement generation

In the algorithm, the list of placements $\mathcal{S}$ is iteratively extended by means of solving consecutive formulations of CPOP of the form $\mathbb{P}[M, \{a^*\}]$ for a particular attack a^* (CPOP is called the *pricing*

problem in this context), and in each iteration, for the current list $\mathcal{S}$, an optimal attack a^* is found by means of solving formulation $\mathbb{A}[K, \mathcal{S}]$ (NAOP is called the *master problem* in this context). 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.

A2: Algorithm for attack optimization by means of controller placement 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: Solve $\mathbb{P}[M, \{a^*\}]$ to get the best placement s^* (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 $\mathbb{A}(\mathcal{S}(M))$, that is $Z^* = V(s(a^*), a^*) = \min_{a \in \mathcal{A}(K)} \max_{s \in \mathcal{S}(M)} V(s, a)$.

5 NUMERICAL EXPERIMENTS

Below we will illustrate the performance of the two algorithms introduced in the previous section. For this purpose we expressed the formulated problems and the algorithms as models and procedures in the AMPL language. We ran the computations on a standard laptop using the AMPL runtime and the CPLEX MIP solver. In our experiment we set the number of controller nodes M to 6 and the number of attacked locations K to 4.

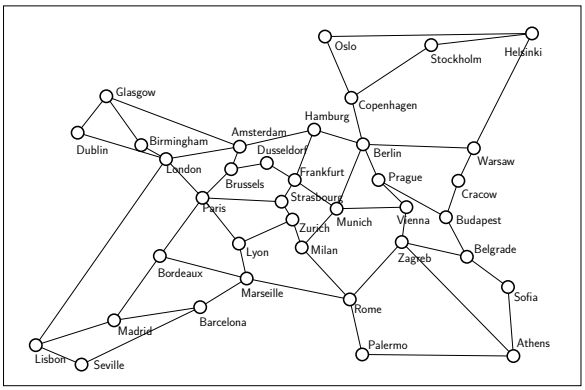

Figure 4: The *cost266* network instance.

The results of applying algorithm A1 to CPOP are summarized in Table 2. In the second row, a medium size network instance *cost66* available in SNDlib [4], whose graph is composed of $V = 37$ nodes and $E = 57$ links, is considered. The optimal objective function value $Y^* = 29$ shows that the optimal placement is capable of assuring that at least 29 nodes out of $V = 37$ nodes will survive any 4-node attack. The optimal solution of $\mathbb{P}[6, \mathcal{A}(4)]$ was reached in 29 iterations of A1, which took 16 seconds of the total computation time on a standard laptop. Most of this time (15 seconds) was spent in Step 1 on solving NAOP. It is remarkable that while the total number of different 4-node attacks equals 66, 045, we needed to generate only 29 of them to get the optimal solution of the considered problem.

The third row of Table 2 shows analogous results for *coronet conus* [8], a network instance substantially larger than *cost66*. This time the optimal placement protects at least $Y^* = 64$ nodes out of 99 nodes for any 4-node attack, and only 72 attacks (out of 3, 764, 376 possible 4-node attacks) need to be generated (which takes only 63 seconds).

Table 2: Results of optimal controller placement.

V	E	M	K	Y^*	$\lvert\mathcal{A}\rvert$	T($\mathbb{P}[6, \mathcal{A}]$)	T($\mathbb{A}[4, \{s^*\}]$)
37	57	6	4	29	29	1 sec.	15 sec.
75	99	6	4	64	72	3 sec.	63 sec.

The results of applying algorithm A2 to NAOP for *cost266* are summarized in Table 3. The optimal objective function value $Z^* = 33$ indicates that the optimal attack is capable of guaranteeing that (only) 4 nodes out of $V = 37$ nodes will be damaged if any of the 6-node controller placements can be deployed. The optimal solution of $\mathbb{A}[4, \mathcal{S}(6)]$ was reached in 40 iterations of A2, which took 1102 seconds in total, and, similarly as for A1, virtually the entire computation time was spent on solving the NAOP problem (this time in Step 2). Again, it is worth noticing that, while the number of different 6-node placements equals 2, 324, 784, we needed to consider only 40 of them to get the optimal solution of $\mathbb{A}[4, \mathcal{S}(6)]$. Table 3 shows no results for *coronet conus* because for this network it took too much computational time to run A2 on the laptop.

Table 3: Results of attack optimization.

V	E	M	K	Z^*	$\lvert\mathcal{S}\rvert$	T($\mathbb{P}[6, \{a^*\}]$)	T($\mathbb{A}[4, \mathcal{S}]$)
37	57	6	4	33	40	1 sec.	1101 sec.

In conclusion, A1 solves the controller placement problem CPOP very quickly, much faster than A2 solves the node attack optimization problem NAOP. So in the case of large networks the efficiency of A2 (which is determined by the efficiency of solving formulation $[K, \mathcal{S}]$) needs to be improved. Fortunately, the limited number of iterations required by A2 will help to achieve this goal.

6 AN ALTERNATIVE RESILIENCE MEASURE

The resilience (to attacks) measure assumed in the previous sections expresses the number of nodes surviving a given attack a. In this section we consider another important measure of this kind, namely the number of surviving (unordered) node-pairs $\{v, w\}$, i.e., the pairs for which both nodes belong to the same component $c \in C(a)$ and this component contains a controller. Note that such a measure is able to account for the traffic relations affected by an attack.

In order to extend the introduced optimization model to the new measure, we need to specify formulations for the counterparts of the two basic problems presented in Section 3. In fact, in the case of CPOP a modified formulation (denoted by $\mathbb{P}'[M, \mathcal{A}]$) is straightforward and merely replaces constraints (3d) with

$$\sum_{v \in \mathcal{V}(c)} y_v^a \leq \binom{|\mathcal{V}(c)|}{2} \sum_{v \in \mathcal{V}(c)} x_v \qquad a \in \mathcal{A},\ c \in C(a) \qquad (9)$$

in the $\mathbb{P}[M, \mathcal{A}]$ formulation (3).

However, in the case of NAOP, modification of formulation $\mathbb{A}[K, \mathcal{S}]$ is not that obvious, and is achieved as described below.

In the modification, apart from variables $a = (a_v)_{a \in \mathcal{V}}, t = (t_e)_{e \in \mathcal{V}}, z = (z_v^s)_{s \in \mathcal{S}, v \in \mathcal{V}}$ and Z, the following additional binary variables are used. For all $v, w \in \mathcal{V}$, let y_{vw} be a binary variable equal to 1 if, and only if, at least one path between nodes v and w composed of links not affected by the constructed attack is available. And for all $s \in \mathcal{S}, v, w \in \mathcal{V}$ and $v < w$, let X_{vw}^s be a binary variable equal to 1 if, and only if, service relation $\{v, w\}$ still provides service after the attack, i.e., there still exists a path between nodes v and w and both v and w are still connected to a controller in placement s (we notice that if the path exists and one of the nodes is connected to a controller, the other node is also connected to a controller). Using these variables the considered modification of the NAOP formulation is as follows:

$$\mathbb{A}'[K, \mathcal{S}] : \min\ Z \qquad (10a)$$

$$\sum_{v \in \mathcal{V}} a_v = K \qquad (10b)$$

$$t_e \geq a_v \qquad v \in \mathcal{V}, e \in \delta(v) \qquad (10c)$$

$$t_e \leq a_{\alpha(e)} + a_{\beta(e)} \qquad e \in \mathcal{E} \qquad (10d)$$

$$z_v^s \leq 1 - a_v \qquad s \in \mathcal{S}, v \in \mathcal{V} \qquad (10e)$$

$$z_v^s \geq 1 - a_v \qquad s \in \mathcal{S}, v \in \mathcal{V}(s) \qquad (10f)$$

$$z_{\alpha(e)}^s \geq z_{\beta(e)}^s - t_e \qquad s \in \mathcal{S}, e \in \mathcal{E} \qquad (10g)$$

$$z_{\beta(e)}^s \geq z_{\alpha(e)}^s - t_e \qquad s \in \mathcal{S}, e \in \mathcal{E} \qquad (10h)$$

$$y_{vw} = y_{wv} \qquad v, w \in \mathcal{V} \qquad (10i)$$

$$y_{vv} = 1 - a_v \qquad v \in \mathcal{V} \qquad (10j)$$

$$y_{v\alpha(e)} \geq y_{v\beta(e)} - t_e \qquad v \in \mathcal{V}(s), e \in \mathcal{E} \qquad (10k)$$

$$y_{v\beta(e)} \geq y_{v\alpha(e)} - t_e \qquad v \in \mathcal{V}(s), e \in \mathcal{E} \qquad (10l)$$

$$X_{vw}^s \geq y_{vw} + z_v^s - 1 \qquad s \in \mathcal{S}, v, w \in \mathcal{V} : v < w \qquad (10m)$$

$$Z \geq \sum_{v,w \in \mathcal{V}: v<w} X_{vw}^s \qquad (10n)$$

$$a_v, t_e, z_v^s \in \{0, 1\} \qquad s \in \mathcal{S}, v \in \mathcal{V}, e \in \mathcal{E} \qquad (10o)$$

$$y_{vw} \in \{0, 1\} \qquad s \in \mathcal{S}, v, w \in \mathcal{V} \qquad (10p)$$

$$X_{vw}^s \in \{0, 1\} \qquad s \in \mathcal{S}, v, w \in \mathcal{V} : v < w \qquad (10q)$$

$$Z \in \mathbb{R}^+. \qquad (10r)$$

In the formulation, constraints (10b)-(10h) imposed on variables a, t and z are the same as constraints (5b)-(5h) in formulation (5). Additional constraints (10i)-(10l), in turn, force that for each node v and each link e unaffected by the constructed attack, node v is either connected (by a path composed of unaffected links) to both ends of the link or is not connected to any of them (i.e., $y_{v\alpha(e)} = y_{v\beta(e)}$ when $t_e = 0$).

Clearly, a necessary and sufficient condition for a pair of nodes $\{v, w\}$ to be in service for a given placement s is that these nodes are connected by means of a path (not necessarily elementary) of surviving links containing an unaffected controller, that is if, and only if, both y_{vw} and z_v^s are equal to 1. To express this condition, constraints (10m) that force variable X_{vw}^s to be equal to 1 only when $y_{vw} = z_v^s = 1$ is introduced.

Finally, for the reasons similar to those used for formulation (5), constraint (10n) together with objective (10a) assure (by setting appropriate values in z_v^s, y_{vw} and X_{vw}^s to 0 when needed) the proper value of the objective function for optimal solutions of the considered formulation.

Clearly, algorithms A1 and A2 formulated in Sections (4.1) and (4.2) remain unchanged when used for the so modified versions of CPOP and NAOP, provided that in the max-min and min-max quantities, respectively, defined at the end of Section 2, the value of $V(s, a)$ expresses the number of surviving node-pairs instead of the number of surviving nodes.

7 FINAL REMARKS

The two problems studied in this paper become important when both the network operator and the attacker are trying to optimize their decisions about, respectively, controller placement and attack selection. In such a case, the presented problems can be of value when the interaction between the two parties is considered within the framework of game theory.

Regarding the direct extensions of the material presented in this paper, we plan to improve the efficiency of the NAOP formulation and thanks to that extend the numerical studies to large networks (e.g., with 100 nodes), also taking into account the alternative resilience measure considered in Section 6.

Finally, let us emphasize that the presented optimization model can be applied to systems other than SDN, mentioned in Section 1.

ACKNOWLEDGEMENTS: This work was supported by the POB Research Centre Cybersecurity and Data Science of Warsaw University of Technology within the Excellence Initiative Program - Research University ID-UB (grant no. CyberiADa/1/2020/W13) and by the National Science Centre, Poland (grant no. 2017/25/B/ST7/02313). At the same time, we thank the anonymous reviewers whose comments helped us improve the presentation.

REFERENCES

[1] Eusebi Calle, David Martínez, Mariusz Mycek, and Michał Pióro. 2021. Resilient backup controller placement in distributed SDN under critical targeted attacks. *Int. J. of Critical Infrastructure Protection* 33 (2021), 12–260.

[2] Tamal Das, Vignesh Sridharan, and Mohan Gurusamy. 2020. A survey on Controller Placement problem in SDN. *IEEE Communications Surveys and Tutorials* 22, 1 (2020), 472–503.

[3] Tao Hu, Zehua Guo, Peng Yi, Thar Baker, and Julong Lan. 2018. Multi-controller Based Software-Defined Networking: A Survey. *IEEE Access* 6 (2018), 15980–15996.

[4] Sebastian Orlowski, Michal Pióro, Artur Tomaszewski, and Roland Wessäly. 2010. SNDlib 1.0 – Survivable network design library. *Networks: An International Journal* 55 (2010), 276–286.

[5] Michał Pióro, Mariusz Mycek, and Artur Tomaszewski. 2021. Network protection against node attacks based on probabilistic availability measures. *IEEE Trans. on Network and Service Management* 18, 3 (2021), 2742–2763.

[6] Dorabella Santos, Amaro de Sousa, Carmen Mas-Machuca, and Jacek Rak. 2021. Assessment of Connectivity-Based Resilience to Attacks Against Multiple Nodes in SDNs. *IEEE Access* 9 (2021), 58266–58286.

[7] Dorabella Santos, Amaro de Sousa, and Paulo Monteiro. 2018. Compact Models for Critical Node Detection in Telecommunication Networks. *Electronic Notes in Discrete Mathematics* 64 (2018), 325–334.

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