A Covering Decomposition Algorithm for Power Grid Cyber-Network Segmentation

Emma S. Johnson[1,2], Santanu S. Dey[1], Jonathan Eckstein[3],
Cynthia A. Phillips[2], John D. Siirola[2]

[1]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA. ejohnson335@gatech.edu, santanu.dey@isye.gatech.edu

[2]Sandia National Laboratories, Albuquerque, NM, USA. {esjohn, caphill, jdsiiro}@sandia.gov

[3]Department of Management Science and Information Systems, Business School Newark and New Brunswick, Rutgers University, Piscataway, NJ, USA jeckstei@business.rutgers.edu

December 13, 2021

Abstract

We present a trilevel interdiction model for optimally segmenting the Supervisory Control and Data Acquisition (SCADA) network controlling an electric power grid. In this formulation, we decide how to partition nodes of the SCADA network in order to minimize the shedding of load from a worst-case cyberattack, assuming that the grid operator has the opportunity for recourse mitigating attack damage. The model is unique in that it couples the cyber and physical networks, using the physical operation of the grid to inform the cyber network segmentation decisions. This feature leads to two violations of assumptions made in much of the prior literature on trilevel interdiction: first, the network designer does not have the ability to make any components of the SCADA or physical networks invulnerable to attack. Instead, the designer makes some attacks more expensive for the attacker. Second, it is possible for the network designer (first player) to make certain attacker (second player) solutions infeasible because he can make them exceed the attacker's budget. In this paper, we present a solution procedure for our formulation that is an adaptation of a covering decomposition algorithm for bilevel interdiction. We show through an empirical study on grids with up to 2,000 buses that this is the first method capable of solving the network segmentation model on realistically sized power grids.

Keywords: trilevel optimization; interdiction; defender-attacker-defender; mixed-integer programming; power networks; security

1 Introduction

This paper describes a trilevel optimization approach to planning the defense of the computer networks controlling electrical power grids. Electrical power grids, like all modern energy distribution infrastructures, are controlled and monitored by SCADA (Supervisory Control and Data Acquisition) systems. These systems are candidates for cyberattacks with potentially severe physical consequences (Genge et al. 2017). Incidents such as the Spring 2021 Colonial Pipeline disruption and the 2015 Ukraine power system cyberattack underscore the likelihood and potential impact of hackers compromising SCADA systems. As the importance of electric power increases in proportion to other forms of energy (e.g., electric vehicles), "smart grid" capabilities expand, and power grid control becomes increasingly networked, electric power grids are becoming increasingly vulnerable and important potential targets. This paper describes a modeling approach and accompanying solution algorithm for mitigating the impact of attacks on power grid control infrastructures.

Our approach does not assume that any part of the control network or power grid infrastructure can be absolutely hardened against assailants. Instead, the goal is to increase the difficulty of an attacker being able to have a significant impact on power delivery. In the model developed in this paper, the basic tool to increase the difficulty of attack is partitioning nodes of the control network into multiple "enclaves," a process called *network segmentation*. Communication among different enclaves is limited, and to control or disable any particular relay in the physical power distribution network, we assume that an attacker must compromise all the control network enclaves to which it is assigned. Each additional enclave that an assailant must compromise to execute an attack is presumed to require time and effort on their part, and also to raise the probability of detection, all of which may be interpreted as costs to the attacker. The general goal of the defender is to configure the enclaves so that an attacker must penetrate a relatively large number of them in order to have a significant effect on the amount of power delivered by the physical network, thus making it harder to execute a high-impact attack. Using network segmentation to mitigate attacks is recommended in U.S. and Australian government documents such as US Department of Homeland Security (2016), Pillitteri and Brewer (2014), Stouffer et al. (2016), and Australian Cyber Security Centre (2020).

This paper presents a model and accompanying solution method for deciding where to best place enclaves in a power grid's SCADA network. The model uses a three-level Stackelberg game framework whose moves proceed as follows:

1. Subject to various budget constraints, the first player, whom we refer to as the *designer*, divides components of the SCADA network into enclaves.

2. The second player, whom we call the *attacker*, then decides which enclaves to compromise and disables the physical elements of the power network controlled by those enclaves. The total number of enclaves compromised is subject to a budget constraint.

3. Finally, by reconfiguring the uncompromised portions of the power network, the third player, whom we call the *defender*, attempts to fulfill as much power demand as is still possible. This process is called "re-dispatch".

The amount of consumer power demand left unmet after the last move is referred to as "load shed". The designer and defender objectives are to minimize load shed and the attacker's objective is to maximize it. Therefore, we are considering an *interdiction* problem, that is, a multilevel optimization problem in which the levels alternate between exactly opposing objectives. Such problems are common in operations research, but in our case the situation is especially challenging because there are three decision levels instead of the more customary two.

Trilevel interdiction formulations of infrastructure security problems have been investigated before, as discussed below in Section 1.1. The model presented here builds on Arguello et al. (2021), where the designer decisions model realistic cybersecurity measures, focusing on specifying the structure of the SCADA network through segmentation. This approach differs from other trilevel formulations where the designer selects particular physical network assets to completely "harden" or make invulnerable to compromise by the attacker. While the trilevel interdiction problem we solve is in essence the same as that proposed by Arguello et al. (2021), we use a different solution procedure that is able to solve larger and more difficult problem instances. To make this procedure more workable, we use a modified formulation with a different attacker strategy representation.

Trilevel models such as we consider here lead to particularly challenging forms of optimization problems. For example, even solving linear bilevel problems is NP-hard (Jeroslow 1985). Here, we take the standard approach of conceptualizing a trilevel problem as a bilevel optimization problem whose second stage is itself a bilevel problem, and then applying various bilevel optimization techniques. Fundamentally, there are three basic approaches to bilevel optimization:

Follower optimality constraints: This approach involves formulating a single-level optimization model that enforces optimality of the follower-controlled variables through explicit constraints. One then formulates the entire bilevel problem as optimization of the leader objective over a feasible region that includes these constraints. For a convex follower problem, the typical approach involves enforcing the Karush-Kuhn-Tucker (KKT) conditions by explicitly including both primal and dual variables for the follower and requiring complementary slackness through disjunctions or auxiliary integer variables combined with "big-M" constraints. In the case of interdiction problems, however, one may be able to take a simpler route in which the follower problem is essentially replaced with its dual, which is solved in conjunction with the leader problem; this approach is known as *dualize and combine*.

Follower optimality by generating cutting planes: This approach starts with the *high-point relaxation*, in which the follower objective is effectively eliminated and leader has full control of the follower variables; both the leader or follower variables may be subject to integrality constraints. While solving the resulting mixed-integer program by branch and cut, one incrementally introduces cutting planes that invalidate solutions that are suboptimal for the follower. This technique is the core of the approach taken in the software of Fischetti et al. (2017) and Tahernejad et al. (2020).

Response-function methods: These techniques, closely related to Benders decomposition, build a progressively more accurate functional model of the follower response to various leader decisions. Each time the model is improved, one computes an optimal leader decision with respect to the current model, finds an optimal follower response, and uses that response to further improve the follower model. Wood (2011) presents such a technique for bilevel interdiction models with convex inner problems. Tang et al. (2016) consider interdiction models with integrality requirements in the inner problem. Lozano and Smith (2017b) propose a similar algorithm for general bilevel problems.

Yue et al. (2019) combine elements of two of the above approaches into a column-and-constraint generation algorithm. As in the follower-optimality-by-cutting-planes approach, the algorithm iteratively refines the high-point relaxation, but — similarly to response-function methods — this refinement is effected by incrementally generating follower solutions. This technique can solve problems in which the leader can make the follower problem infeasible, unlike the branch-and-cut techniques mentioned above. We also refer the interested reader to Smith and Song (2020) for a survey of algorithms for network interdiction problems.

In our trilevel model, the third stage is a continuous, convex power flow problem for which we use a network flow relaxation, an approximation that we justify through the results in Johnson and Dey (2021); see Section 3.2.2 for a more detailed discussion. The resulting convexity of the last stage makes it natural to use an approach in the follower-optimality-constraint class to link the second and third stages. Since we have an interdiction model, we are able to use the dualize-and-combine technique to merge the second and third stages, essentially replacing the re-dispatch problem with its dual, solved in conjunction with the attack decisions. We call the resulting model the "attacker MIP" (since the attacker's decisions are discrete).

Using this kind of merger of the second and third stages, Arguello et al. (2021) attempt to solve the full trilevel problem by applying the follower-optimality-by-cutting-planes method, as embodied in the software of Fischetti et al. (2017), to the mixed-integer bilevel problem whose first stage consists of the network segmentation decisions and whose second stage is the attacker MIP. However, this approach has difficulty solving instances of realistic size, likely because the high-point relaxation used by the cutting plane approach is particularly weak for interdiction problems: since the leader and follower have opposite objectives, the high-point relaxation chooses solutions that are very far from optimal for the follower, and large numbers of cutting planes must be incrementally added to guide the solution toward follower optimality. These difficulties agree with the observations regarding the unsuitability of general bilevel methods for interdiction problems made in Wood (2011) and Hua et al. (2019); the latter state that "relaxations based on the high-point problem are weak for an interdiction problem with its directly conflicting objective functions."

Here, we similarly use dualize and combine to link the second and third stages, but coordinate the first and second stages differently, using a specialized response-function technique. The result may be viewed as a trilevel adaptation of the covering decomposition method proposed for bilevel problems by Israeli and Wood (2002). Our algorithm maintains a set of anticipated attacks K that might be executed by the assailant, and uses the following basic loop:

1. Attempt to compute a defender configuration that "blocks" all the attacks in K, that is, makes them impossible to execute within the attacker budget. This step involves solving a feasibility problem, which in our case involves only binary variables.

2. Compute the highest-impact attack k possible on the defender configuration just computed in step 1.

3. Update the set of anticipated attacks K to include k, and return to step 1.

If there is no solution to the feasibility problem in step 1, then the algorithm terminates with the optimal defense configuration being the the best one generated so far. The intention of the method is that termination may be achieved with K containing only a tiny fraction of all possible attacks.

Our computational experiments (see Section 4), demonstrate that our method can find optimal solutions to problems on realistically sized power grids with up to 2,000 buses, so long as the attacker and defender budgets are relatively small (a "bus" is essentially equivalent to a node in the power network). For smaller grids, we can find optimal solutions for larger budgets.

The computational results below, which contain a grid of designer and attacker budget combinations, also suggest how the network designer might assess the impact of various designer budgets on network security. Such an assessment could then eventually lead to the selection of a particular budget and a concrete network segmentation plan.

While our approach effectively solves the same class of problem as in Arguello et al. (2021), using a response-function methodology to link the first and second stages necessitated a new, different formulation. Specifically, we do not describe potential attacks by which enclaves they penetrate, as in Arguello et al. (2021). Instead, each attack is effectively a list of the physical power network elements the attacker needs to control, from which the required set of enclave penetrations can be deduced for any particular control network configuration. This change obviates numerous potential model symmetries and makes a response-function approach practicable.

In summary, this paper makes three closely interrelated contributions to the defense of power-grid SCADA networks: (1) reformulation of the general approach proposed in Arguello et al. (2021) to make a response-function solution method more practical, (2) application of a response-function method, specifically an adaptation of the covering decomposition algorithm of Israeli and Wood (2002) to trilevel problems in power-grid security, and (3) the first demonstrated solution of trilevel control network segmentation models derived from realistically sized power grids. Since network segmentation formulations model power grid cybersecurity much more realistically than previous physical-component-hardening models, being able to solve network segmentation problems for realistically sized power grids is a significant advance.

1.1 Literature Review

There has been much interest in trilevel interdiction in the past twenty years, particularly with applications in power systems resilience (Oster et al. 2020). However, the field appears to contain relatively few truly distinct algorithmic approaches. In general, these ideas can be sorted into two broad categories: for trilevel interdiction problems where the second player's feasible region does not depend on the first player's decision, methodology from two-stage robust optimization applies. For problems where this independence property does not hold, several techniques originally developed for bilevel problems apply, most often the covering decomposition method from Israeli and Wood (2002) and the implicit enumeration approach from Scaparra and Church (2008). For problems where the third-player problem is convex, it is common to reformulate the trilevel problem as a bilevel problem by using dualize and combine on the inner two levels, after which bilevel solution algorithms can be applied directly. Otherwise, bilevel decomposition algorithms can be applied in a nested fashion. Solving the subproblem then also requires an algorithm for bilevel optimization. In the remainder of this section, we survey some trilevel interdiction problems in a variety of domains, with a focus on the methodology used to solve them. When referring to upper and lower bounds on the solution value, we will use the convention that all problems are in a min-max-min format, as in our formulation. For problems originally having a max-min-max format, we thus reverse the meaning of "upper" and "lower" from the original publication.

Numerous papers formulate trilevel interdiction problems so that the feasible region of the second player does not depend on the first player's decision. The resulting formulations have the same basic structure as two-stage robust optimization problems, allowing the application of algorithms from the robust optimization literature. In this context, many authors have had success applying the column-and-constraint-generation algorithm from Zeng and Zhao (2013). For example, Yuan et al. (2014) apply this algorithm to a defender-attacker-defender problem to decide which lines in a transmission grid to harden against attack. Ding et al. (2018) extend this formulation to handle multiple uncertainty sets for the attacker and again apply the Zeng and Zhao column-

and-constraint-generation algorithm. Fang and Zio (2017) use the same algorithm to solve an infrastructure resilience defender-attacker-defender model, and Wu and Conejo (2017) use it to solve an electric grid defense planning problem. Lai et al. (2019) use the same approach to solve a trilevel formulation for allocating resources to defend against a coordinated cyber and physical attack on the power grid. None of these methods are directly applicable to our network segmentation formulation since they cannot handle feasibility dependence between the first two players.

Alderson et al. (2011) develop an algorithm for defender-attacker-defender problems which may also be viewed as a column-and-constraint-generation algorithm. They assume that the defender can make particular physical-grid assets impenetrable to the attacker. Doing so does not make the attacker problem infeasible, but instead means that the attacker gets no benefit from targeting impenetrable assets. This means that, again, the second-player feasible region does not depend on the first-player decision. Under these assumptions, Alderson et al. then present a similar algorithm to Zeng and Zhao (2013), generating attacks to add to a master problem that provides a lower bound. The damage from the attacks gives an upper bound. This approach is very similar to the algorithm presented in our paper, but it is not directly applicable since our model's first player can make particular attacks infeasible, a property that adds disjunctions to the lower-bounding problem. Furthermore, we do not assume that the first player can make physical assets invulnerable. Ouyang (2017), Ouyang and Fang (2017), Ouyang et al. (2017), and Ouyang et al. (2018) all adapt the algorithm of Alderson et al. (2011) to various trilevel problems to guide investments to enhance resilience of critical infrastructure.

For trilevel interdiction problems with a convex third level, it is common to use a dualize-and-combine reformulation to transform the inner bilevel problem into a single-level problem. For example, Alguacil et al. (2014) consider a power-grid hardening problem taking into account a set of plausible contingencies. Similarly to Alderson et al. (2011), their model assumes that the defender can make some network assets impenetrable. The third-level (defender) problem is a DC optimal power flow (DCOPF) problem, which is convex, so they may use dualize and combine to create a bilevel problem to which they directly apply the implicit enumeration algorithm of Scaparra and Church (2008). While a variation on this method would be possible in principle for the network segmentation problem, the decision space to enumerate would be much larger because one cannot directly fortify physical network components.

Our paper is not the first to adapt the covering decomposition algorithm from Israeli and Wood (2002) to a trilevel setting. Yao et al. (2007) consider a trilevel problem to defend a power network against terrorist attacks. They apply the covering decomposition algorithm to the trilevel problem in a nested fashion: they treat the first-player problem as a feasibility problem, and again use the covering decomposition algorithm in order to find the worst-case attack solutions to the inner bilevel problem. The covering decomposition algorithm is also applied to a protection-interdiction-restoration model for grid resilience in Ghorbani-Renani et al. (2020), but this methodology again depends on being able to directly protect individual physical assets. Similarly to Yao et al. (2007), they use covering decomposition in a nested fashion, although their first-player problem has an objective function.

Ghorbani-Renani et al. (2021) present a general algorithm for trilevel optimization, which they decompose into two bilevel problems: a master problem in which the second player solution is fixed, giving a lower bound, and a subproblem where the first-player solution is fixed, giving an upper bound. These bounds are closed by alternating between the two problems and adding cuts to the subproblem requiring that it does not encounter the same solution twice. This kind of approach is

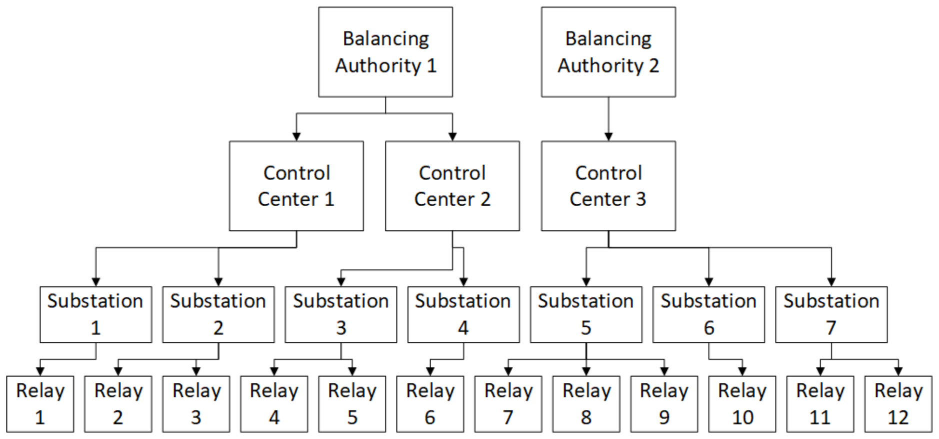


Figure 1: Control network structure. Balancing authorities enclaves communicate with control center enclaves which in turn communicate with substation enclaves. These then control the relays, each of which controls a set of physical grid components (i.e., lines, generators, and loads).

again only valid if the first player cannot make second-player solutions infeasible.

Lozano and Smith (2017a) present a novel backward-sampling algorithm for trilevel interdiction problems in which the sets of assets that can be protected and attacked are the same, and the previously mentioned assumption about impenetrable assets holds. This assumption allows the feasible region of the second problem to be independent of the first-player problem decision. The first two problems must be pure binary, while the third problem may have any mix of integer and continuous variables. The algorithm is based on iteratively sampling from the third player's feasible solutions, with each sample providing a restriction of the trilevel problem. This restriction is used to construct both upper and lower bounds and to identify assets that the first player must defend. However, this method is not directly applicable to our network segmentation model, since we do not assume that the first player can make some assets impenetrable. In addition, because the inner two problems of our model are easily reformulated as a single-level problem, we do not need the backward sampling methodology.

2 Network Segmentation Model

We now present a novel formulation of the network segmentation problem first developed in Arguello et al. (2021), with several modifications. We model the SCADA system controlling the transmission grid as a forest with depth four. *Balancing authorities* are the root nodes, and their descendants (in order) are *control centers*, *substations*, and *relays*. The relays control the lines, generators, and loads on the physical grid. Throughout this paper, substations are in one-to-one correspondence with the buses on the physical grid. Figure 1 shows a diagram of an example control network.

The goal of the network segmentation problem is to divide selected balancing authority, control center, and substation nodes into multiple cybernetwork partitions, henceforth referred to as *enclaves*, so as to contain the load shed from a worst-case cyberattack as much as possible. Unpartitioned nodes of the original network constitute a single enclave, and the number of new enclaves that may be introduced is subject to a budget constraint.

We formulate the network segmentation problem as a trilevel optimization. In the first-player

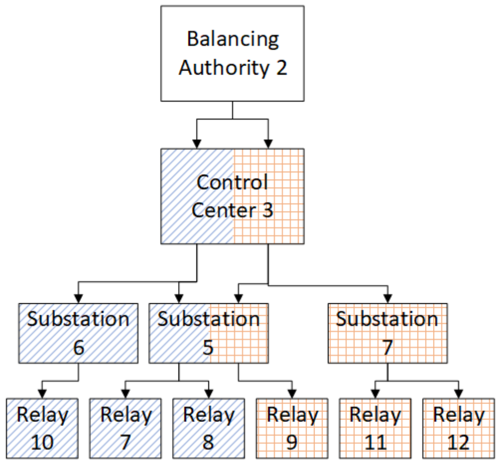


Figure 2: Example of a segmented portion of the control network from Figure 1. The balancing authority still contains one enclave, but the control center and substation 5 now contain two enclaves each. The substations and relays are patterned according to which control center enclave ultimately controls them. The enclaves in substation 5 are controlled by a different parents.

problem, the designer decides where to add new enclaves subject to two separate budget cardinality constraints, one for the balancing authorities and control centers combined, and one for the substations. If no enclaves are added to a node, it contains only its single, original enclave. The designer also determines parent-child relationships between the enclaves, respecting the original control hierarchy: given a node n with parent $P(n)$ in the original control network, each enclave within n must be the child of a single enclave within $P(n)$, but the designer is free to choose which one whenever $P(n)$ has multiple enclaves. The resulting, segmented control network remains a forest. For example, Figure 2 shows a segmented version of the "balancing authority 2" tree from Figure 1: here, no enclaves were added to balancing authority 2, but control center 3 and substation 5 each have one new enclave. Each of the nodes is patterned according to which control center network ultimately controls it. Note that the partition refines the original forest structure: for example, relays 7, 8, and 9 are still children of substation 5 and grandchildren of control center 3, but relay 9 is now in different control center and substation enclaves than relays 7 and 8. Figure 3a and Figure 3b show two examples of disallowed network segmentation decisions: in the former, relay 8 is the child of two different enclaves within substation 5, while in the latter, relay 7 has become a child of an enclave in substation 6 despite originally being a child of substation 7 (Figure 3c violates a different constraint that we will describe below).

After the designer has selected a control network segmentation scheme, the attacker maximizes load shed by compromising relays and shutting off the grid components they control. To gain access to a relay, the attacker must have access to the substation enclave that controls it. In turn, to obtain access to any enclave, he must have access to its parent. Thus, the attacker is selecting a subforest of the post-segmentation control network graph that maximizes the load shed (after the defender's response) when he controls its leaves. The attacker is assumed to be subject to a limit on a number of enclaves (not including relays) that he can compromise. For example, with a budget of three enclaves in the unsegmented network in Figure 1, the attacker might choose to compromise balancing authority 2, control center 3, and substation 5, hence gaining control of relays 7, 8, and

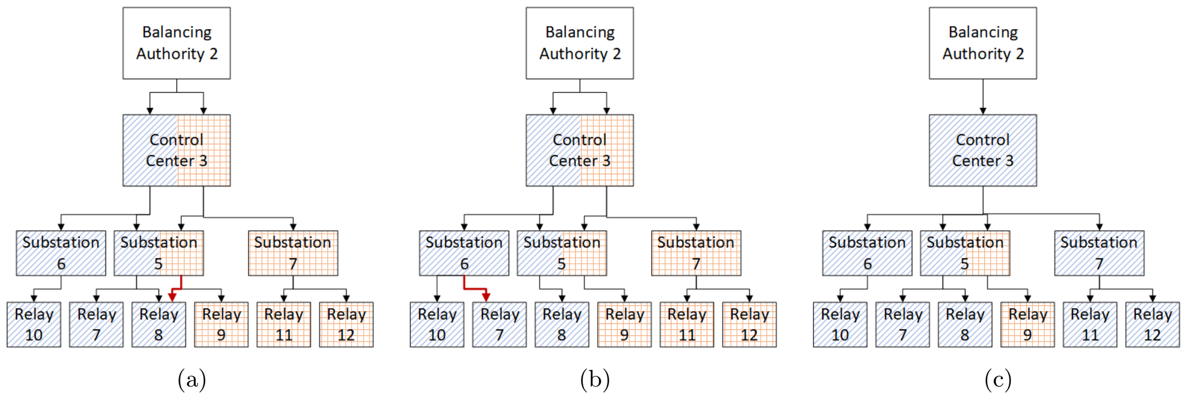


Figure 3: Examples of disallowed network segmentation choices. In (a), relay 8 has been assigned to both of the enclaves in substation 5, and the resulting network is no longer a forest. In (b), the original network structure is violated since relay 7 was originally a child of substation 5, but has been reassigned to substation 6. In (c), substation 5 is segmented, but both of its enclaves are assigned to the same control center enclave (see below for this restriction).

9. In the segmented network partially displayed in Figure 2, attacking the same set of relays would require a budget of five enclaves, since the attacker would need to compromise both enclaves in control center 3 and both enclaves in substation 5.

Last, in the third-player problem, the defender re-dispatches the uncompromised portions of the grid, as modeled by a capacitated network flow problem. Taken together, the entire formulation is therefore a trilevel interdiction problem with pure binary first and second actors and a convex third level. For further detail on the model, we refer the reader to Arguello et al. (2021).

We enhance the realism of the model from Arguello et al. (2021) with the additional constraint that the network designer cannot assign multiple substation enclaves within a single substation to be controlled by the same control center enclave. Figure 3c shows a variation of the segmentation from Figure 2 that violates this constraint: it assigns both enclaves in substation 5 to the blue (striped) enclave in control center 3. The reasoning behind this constraint is that workstations and thus network access points are located at the control center level of the SCADA network. We thus assume that an attacker will scan the network from the control center level. There is thus no point in having multiple substation enclaves controlled by the same control center enclave, since both would be fully visible to an attacker having penetrated that enclave. Note, however, that the attacker still pays a unit of budget to pivot into a substation enclave, as this action still increases his risk of detection. Last, Arguello et al. (2021) use three separate budgets in the designer problem, treating balancing authorities and control centers separately. We combine these two budgets, allowing the optimization model to resolve the trade-offs between them.

In Figure 4, we show the control network and physical network for a segmented 30-bus system. Due to space constraints, we do not depict the relays in the communication network, so the displayed leaf nodes correspond to the substations on the physical grid, which correspond to buses in the power network.

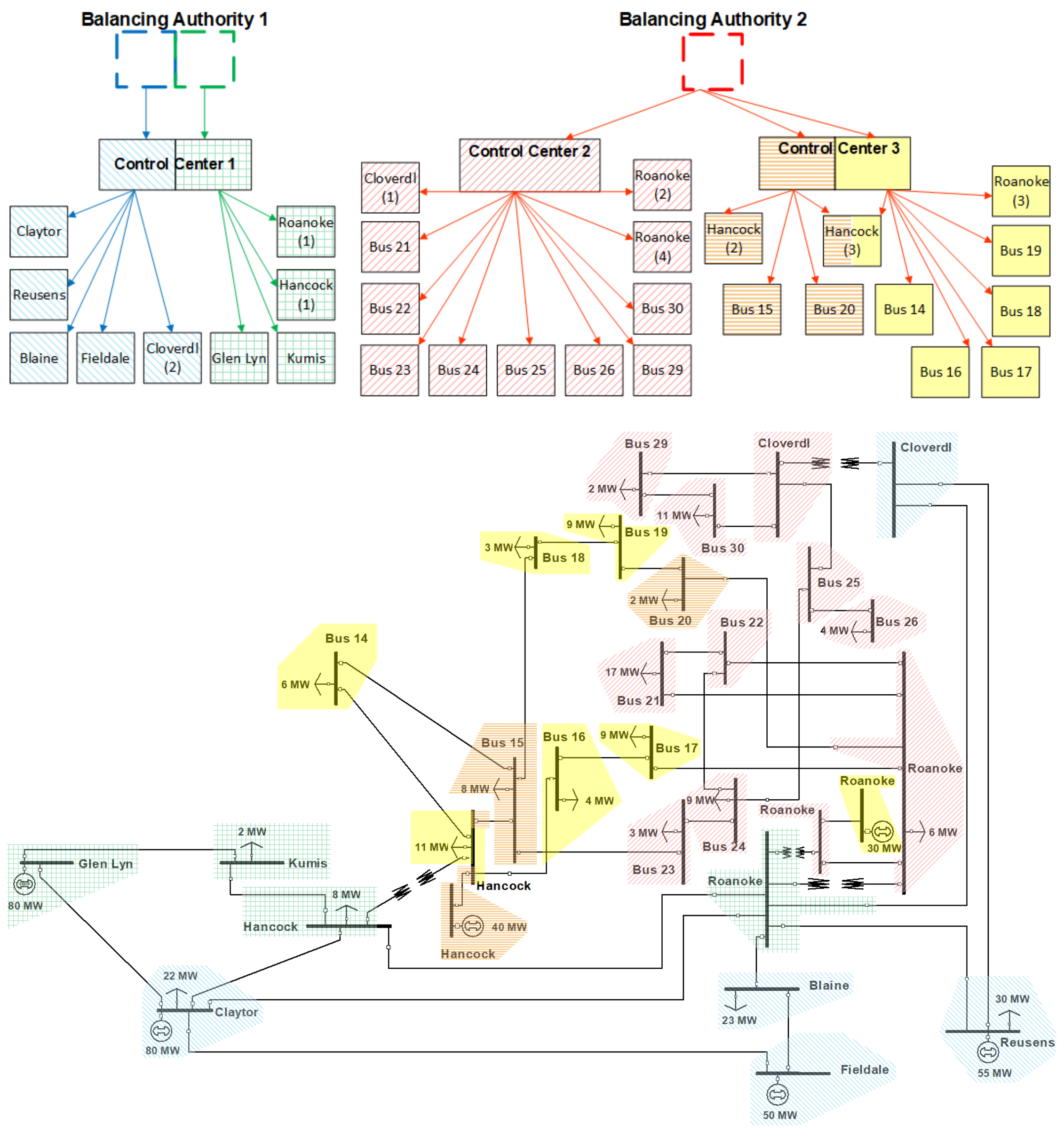


Figure 4: How segmentation of a 30-bus system's control network relates to the physical network. The top image displays the segmented control forest, and the lower image shows its relation to the physical grid. Each leaf node in the forest corresponds to one substation/bus on the physical grid. Substations are colored according to their parent control center enclaves.

2.1 Formulation Nomenclature

2.1.1 Sets

$\mathcal{G}$ Set of generators

$\mathcal{R}$ Set of relays

$\mathcal{B}$ Set of balancing authorities

$\mathcal{C}$ Set of control centers

$\mathcal{S}$ Set of substations

$\mathcal{N}$ Set of all nodes in communication network: $\mathcal{N} = \mathcal{B} \cup \mathcal{C} \cup \mathcal{S}$

$\overline{K}$ Set of attacks on the communication network (defined by the sets of relays they compromise)

$\mathcal{L}$ Set of all transmission lines

$\mathcal{L}_s^+$ Set of transmission lines whose destination is substation $s \in \mathcal{S}$

$\mathcal{L}_s^-$ Set of transmission lines originating at substation $s \in \mathcal{S}$

$\mathcal{G}_s$ Set of generators at substation $s \in \mathcal{S}$

$\mathcal{R}_l$ Set of relays which control transmission line $l \in \mathcal{L}$

$\mathcal{R}_g$ Set of relays which control generator $g \in \mathcal{G}$

$\mathcal{R}_s$ Set of relays which control load at substation $s \in \mathcal{S}$

$E(n)$ Set of possible enclaves in node $n \in \mathcal{N}$

$C(n)$ Set of child nodes of node $n \in \mathcal{N}$

$R(n)$ Set of relays that are descendants of node $n \in \mathcal{N}$. Note that $R(n) = C(n)$ for $n \in \mathcal{S}$

$A(k)$ Relays compromised in attack $k \in \overline{K}$

2.1.2 Parameters

$P(n)$ Parent of node $n \in \mathcal{N}$

D_s Demand at substation $s \in \mathcal{S}$

$\overline{F}_l$ Thermal limit of line $l \in \mathcal{L}$

$\overline{P}_g$ Maximum dispatch of generator $g \in \mathcal{G}$

T Maximum number of balancing authority and control center enclaves network designer can add to the control network

V Maximum number of substation enclaves network designer can add to the control network

W Maximum number of enclaves that the attacker can compromise

2.1.3 Decision variables

We formulate the designer problem as if every node in the communication network has a number of potential enclaves equal to the budget at its tier of the network, plus one; specifically we use the indexing scheme that $E(n) = \{0, 1, \ldots, T\}$ for all $n \in \mathcal{B} \cup \mathcal{C}$. The designer decides which of these enclaves to "activate," that is, add to the network. We do not explicitly model the enclaves at the substation tier of the network. Instead, we assign relays to be children of control center enclaves, which determines the enclaves at the substation level because of the constraint that no control center enclave has more than one child enclave in the same substation. In the following description, a variable defined as "whether or not" some statement holds is binary, being 1 if the statement is true and 0 if it is false. The designer problem variables are as follows:

$x_{n,i}$ Whether or not enclave i in node n is active (added to the network)

$y_{n,i,j}$ Whether or not enclave j in node $P(n)$ is the parent of enclave i in node n

$q_{r,j}$ Whether or not relay r is controlled by enclave j in control center $P(P(r))$

$z_{s,i}$ Whether or not any relay belonging to substation s is a child of enclave i in control center $P(s)$.

The attacker problem variables are:

δ_r Whether or not relay r is compromised

$\alpha_{n,i}$ Whether or not enclave i in node n is compromised

$\beta_{s,i}$ Whether or not to compromise the enclave in substation s that is controlled by enclave i in control center $P(s)$

u_s Whether or not load at substation s is compromised, that is, all the load is shed

v_l Whether or not transmission line l is compromised, that is, its flow is forced to zero

w_g Whether or not generator g is compromised, that is, its power generation is forced to zero.

One may consider u, v, and w as being determined by the choice of δ, since we assume the attacker compromises all physical components he has access to. Last, the defender problem variables are:

f_l Directed flow on line l

p_g Power generated by generator g

ℓ_s Load shed at substation s

2.2 Formulation

Our network segmentation problem is given by

$$\min_{(x,y,q,z)\in\mathcal{D}} \left\{ \max_{(\alpha,\beta,\delta,u,v,w)\in\mathcal{A}(y,q)} \left\{ \min_{(p,f,l)\in\mathcal{X}(u,v,w)} \left\{ \sum_{s\in\mathcal{S}} \ell_s \right\} \right\} \right\} \tag{1}$$

where the set $\mathcal{D}$ is defined by:

$$
\begin{aligned}
& y_{n,i,j} \leq x_{P(n),j} && \forall n \in \mathcal{C}, i \in E(n), j \in E(P(n)) && (2)\\
& q_{r,j} \leq x_{P(P(r)),j} && \forall r \in \mathcal{R}, j \in E(P(P(r))) && (3)\\
& \sum_{j \in E(P(n))} y_{n,i,j} = x_{n,i} && \forall n \in \mathcal{C}, i \in E(n) && (4)\\
& \sum_{j \in E(P(P(r)))} q_{r,j} = 1 && \forall r \in \mathcal{R} && (5)\\
& x_{n,i} \leq \sum_{v \in C(n)} \sum_{j \in E(v)} y_{v,j,i} && \forall n \in \mathcal{B}, i \in E(n) && (6)\\
& x_{n,i} \leq \sum_{r \in R(n)} q_{r,i} && \forall n \in \mathcal{C}, i \in E(n) && (7)\\
& z_{s,i} \geq q_{r,i} && \forall s \in \mathcal{S}, r \in C(s), i \in E(P(S)) && (8)\\
& \sum_{n \in \mathcal{B} \cap \mathcal{C}} \sum_{i \in E(n) \setminus \{1\}} x_{n,i} \leq T && && (9)\\
& \sum_{n \in \mathcal{S}} \sum_{i \in E(P(s))} z_{s,i} \leq V && && (10)\\
& x_{n,i} \in \{0,1\} && \forall n \in \mathcal{B} \cup \mathcal{C}, i \in E(n) && (11)\\
& y_{n,i,j} \in \{0,1\} && \forall n \in \mathcal{C}, i \in E(n), y \in E(P(n)) && (12)\\
& q_{r,j} \in \{0,1\} && \forall r \in \mathcal{R}, j \in E(P(r)) && (13)
\end{aligned}
$$

Recall from Section 2.1.3 that $x_{n,i}$ indicates whether enclave i in node n is active and $y_{n,i,j}$ indicates whether enclave i in node n is a child of enclave j in node $P(n)$. Keeping these definitions in mind, the constraints (2) enforce that enclave j in the parent balancing authority $P(n)$ of some control center n must be active if it has been designated to be the parent of some enclave i at n. The constraints (3) require if relay r is controlled (indirectly through some substation) by enclave j in control center $P(P(r))$, then that enclave must be active. Next, the constraints (4) enforce that active enclaves have exactly one parent enclave and inactive enclaves have no parent. Similarly, the constraints (5) require every relay have exactly one parent enclave. The constraints (6) require that if an enclave is designated as active, it must have at least one active child. Similarly, constraints (7) enforce that active control center enclaves have at least one grandchild relay. For each substation $s \in \mathcal{S}$ and enclave i in control center $P(s)$, the constraints (8) effectively define $z_{s,i}$ as an indicator of whether enclave i is a grandparent of some relay controlled by s. Constraint (9) enforces the investment budget governing how many enclaves can be added to balancing authorities and control centers, combined. Constraint (10) enforces the investment budgets regarding how many new enclaves can be added to substations. While the above constraints are enough to define the problem, we also include some constraints to reduce symmetry, namely

$$x_{n,i-1} \geq x_{n,i} \quad \forall n \in \mathcal{B} \cup \mathcal{C}, i \in E(N) \setminus \{0\}, \tag{14}$$

which requires that the active enclaves in control centers and balancing authorities be numbered sequentially starting at 0.

Given some fixed designer solution $\hat{y}$ and $\hat{q}$, the attacker's feasible region $\mathcal{A}(\hat{y}, \hat{q})$ is defined by the constraints:

$$\delta_r \leq \sum_{i \in E(P(P(r)))} \hat{q}_{r,i} \alpha_{P(P(r)),i} \qquad \forall r \in \mathcal{R} \tag{15}$$

$$\alpha_{n,i} \leq \sum_{j \in E(P(n))} \hat{y}_{n,i,j} \alpha_{P(n),j} \qquad \forall n \in \mathcal{C}, i \in E(n) \tag{16}$$

$$\beta_{s,i} \geq \hat{q}_{r,i} \delta_r \qquad \forall s \in \mathcal{S}, i \in E(P(s)), r \in C(s) \tag{17}$$

$$\hat{q}_{r,i} \beta_{s,i} \leq \delta_r \qquad \forall s \in \mathcal{S}, i \in E(P(s)), r \in C(s) \tag{18}$$

$$\sum_{s \in \mathcal{S}} \sum_{i \in E(P(s))} \beta_{s,i} + \sum_{n \in \mathcal{B} \cup \mathcal{C}} \sum_{i \in E(n)} \alpha_{n,i} \leq W \tag{19}$$

$$v_l \leq (1 - \delta_r) \qquad \forall l \in \mathcal{L}, r \in \mathcal{R}_l \tag{20}$$

$$v_l \geq \sum_{r \in \mathcal{R}_l} (1 - \delta_r) - |\mathcal{R}_l| + 1 \qquad \forall l \in \mathcal{L} \tag{21}$$

$$w_g \leq (1 - \delta_r) \qquad \forall g \in \mathcal{G}, r \in \mathcal{R}_g \tag{22}$$

$$w_g \geq \sum_{r \in \mathcal{R}_g} (1 - \delta_r) - |\mathcal{R}_g| + 1 \qquad \forall g \in \mathcal{G} \tag{23}$$

$$u_s \leq (1 - \delta_r) \qquad \forall s \in \mathcal{S}, r \in \mathcal{R}_s \tag{24}$$

$$u_s \geq \sum_{r \in \mathcal{R}_s} (1 - \delta_r) - |\mathcal{R}_s| + 1 \qquad \forall s \in \mathcal{S} \tag{25}$$

Recall that δ_r indicates whether the attacker compromises relay r, while $\alpha_{n,i}$ indicates whether he must attack enclave i in node n. Thus, the constraints (15) mean that in order for the attacker to compromise relay r, he must also compromise the control center enclave that is its ancestor. The constraints (16) require that to compromise an enclave in control center $n \in \mathcal{C}$, the attacker must compromise its parent enclave in the parent balancing authority $P(n)$. The variable $\beta_{s,i}$ denotes whether or not the attacker compromises the enclave in the substation s that is the child of enclave i in the substation's parent control center $P(s)$. The constraints (17) thus require that if any relay r that is a child of substation s and a descendant of enclave i in control center $P(s)$ is attacked, then the corresponding enclave in s must be attacked as well. The constraints (18) require that the attacker must attack all the relays he has access to by mandating that whenever he acquires access to some substation enclave, he must attack all its children. The attacker's budget is enforced by constraint (19). The constraints (20)-(25) translate between the set of relays compromised by the attack and the set of physical network components available after the attack. For example, the constraints (20) enforce that if a relay controlling line l is compromised, then line l is turned off. Conversely, constraints (21) enforce that whenever line l is off, at least one relay that controls it must be attacked.

Last, given fixed attacker decisions $\hat{u}$, $\hat{v}$, and $\hat{w}$, the defender's feasible region $\mathcal{X}(\hat{u}, \hat{v}, \hat{w})$ is defined by the constraints:

$$\sum_{l \in \mathcal{L}_s^+} f_l - \sum_{l \in \mathcal{L}_s^-} f_l + \sum_{g \in \mathcal{G}_s} p_g + \ell_s = D_s \qquad \forall s \in \mathcal{S} \tag{26}$$

$$D_s(1 - \hat{w}_s) \leq \ell_s \leq D_s \qquad \forall s \in \mathcal{S} \tag{27}$$

$$-\overline{F}_l \hat{v}_l \le f_l \le \overline{F}_l \hat{v}_l \qquad \forall l \in \mathcal{L} \tag{28}$$
$$0 \le p_g \le \overline{P}_g \hat{u}_g \qquad \forall g \in \mathcal{G} \tag{29}$$
$$\ell_s \ge 0 \qquad \forall s \in \mathcal{S} \tag{30}$$

Flow balance at each substation is enforced by the constraints (26), where f_l is the directed flow on line l, p_g is the power generated at generator g, and ℓ_s is the load shed at substation s. The constraints (27)-(30) enforce variable bounds for the components still online after the attack, forcing all attacked components to be off.

3 Methodology

We solve the network segmentation model with an adaptation of the covering decomposition algorithm from Israeli and Wood (2002). We present our method in the form of two equivalent algorithms for solving a generic trilevel interdiction problem, and then discuss their application to our network segmentation problem (1) in Section 3.2. The general trilevel interdiction problem we consider is of the form

$$\nu := \min_{d \in \mathcal{D}} \left\{ \max_{a \in \mathcal{A}(d)} \left\{ \min_{x \in \mathcal{X}(a)} f(x) \right\} \right\}, \tag{31}$$

subject to the following assumptions:

Assumption 1. *We have relatively complete recourse at every level: That is, given $d \in \mathcal{D}$, one has $\mathcal{A}(d) \neq \emptyset$. Similarly, given any $d \in \mathcal{D}$ and $a \in \mathcal{A}(d)$, one has $\mathcal{X}(a) \neq \emptyset$.*

Assumption 2. *The set $\mathcal{D}$ is finite, as is the set $\mathcal{A}(d)$ for all $d \in \mathcal{D}$.*

Assumption 3. *The function f is bounded below by some number $\underline{u}$.*

While the relatively complete recourse assumption specifies that $\mathcal{A}(d)$ is nonempty for any $d \in \mathcal{D}$, it should be kept in mind that $\mathcal{A}(d)$ depends on d, so the designer may have the ability to make any particular attack a infeasible. Let

$$\{a^1, a^2, \ldots, a^H\} = \bigcup_{d \in \mathcal{D}} \mathcal{A}(d) \tag{32}$$

and $\overline{K} = \{1, 2, \ldots, H\}$ be the corresponding index set for all possible attacks. Without loss of generality, we assume that there exists some fixed integer N such that the set $\mathcal{X}(a^k) \subseteq \mathbb{R}^N$ for all $k \in \overline{K}$. Then we may equivalently write (31) as

$$\nu = \min_{\substack{d \in \mathcal{D} \\ \xi \in \mathbb{R} \\ \eta \in \mathbb{R}^H \\ x^1,\ldots,x^H \in \mathbb{R}^N}} \quad \xi \tag{33}$$

$$\text{s.t.} \quad \xi \ge \eta_k \qquad \forall k \in \overline{K} \tag{34}$$

$$\begin{bmatrix} \eta_k = f(x^k) \\ x^k \in \mathcal{X}(a^k) \\ a^k \in \mathcal{A}(d) \end{bmatrix} \vee \begin{bmatrix} \eta_k = \underline{u} \\ a^k \notin \mathcal{A}(d) \end{bmatrix} \qquad \forall k \in \overline{K}. \tag{35}$$

By Assumption 2, $H < \infty$. Also by Assumption 2 and because the attacker problem is discrete, the constraints to require $a^k \notin \mathcal{A}(d)$ in the second disjunct in (35) define a discrete and finite set. Thus, it is possible at least in principle to solve (33)-(35), for example by introducing some additional variables to describe the disjunctions and expressing the problem as a mixed-integer program.

The key idea of this formulation is that by enumerating the second players's feasible set, we are able to combine the minimization operations of the first and third problems. The disjunctions in (35) represent the choice, for each second-player solution, to either find a third-player response (in the first disjunct), or for the first player to block the solution entirely (in the second disjunct). This disjunction is a consequence of the dependence between the second player's feasible region $\mathcal{A}(d)$ on the first player's decision d. The third-player response x^k will be optimal for k where (34) is tight, since ξ is being minimized and does not appear in any other constraints. Thus, d will be an optimal solution to (31).

Next, consider replacing $\overline{K}$ with $K \subset \overline{K}$ in the formulation (33)-(35). Since this change removes some of the disjunctive constraints, the modified problem is a relaxation of the original one and provides a lower bound on ν. Incrementally adding attacks to K means incrementally adding disjunctive constraints, so it will result in a series of progressively tighter relaxations and hence monotonically nondecreasing lower bounds.

For practical applications, H will clearly be very large, meaning that (33)-(35) would have intractable numbers of constraints and variables. Instead of solving (33)-(35) directly, we propose an iterative algorithm that incrementally generates members of the set $\{a^1, a^2, \ldots, a^H\}$. This algorithm relies on being able to solve the bilevel problem formed by the inner two problems as a function of d:

$$u(d) = \max_{a \in \mathcal{A}(d)} \left\{ \min_{x \in \mathcal{X}(a)} f(x) \right\}. \tag{36}$$

While this problem may also be difficult, there are cases in which it may be tractable: for example, if the third-player problem is convex, then dualize-and-combine methods or the global Benders approach described in Wood (2011) are applicable.

For any feasible first-player solution d, the value $u(d)$ defined in (36) is an upper bound on the optimal objective value since it is the objective value of the second player's best response to a feasible first-player decision. The main idea of the algorithm is to alternate between solving (36) to get a second-player solution a^k, and thus an upper bound, and solving a relaxation of (33)-(35) that replaces $\overline{K}$ with the set K of indices of the second-player solutions enumerated so far. This relaxation of the original problem yields a lower bound and a new first-player solution to fix in (36). The details of the algorithm are given in Algorithm 1.

Based on our assumptions, Algorithm 1 terminates in finitely many iterations. Formally, we have the following result:

Proposition 1. *Under Assumptions 1-3, Algorithm 1 converges in finite time. The designer solution d^* it returns is ϵ-optimal for* (31)*, in the sense that its objective value U is no more than $\nu + \epsilon$.*

Proof. We claim that the algorithm cannot generate the same attack a^k twice, except perhaps in the final iteration. To see this, suppose that $0 \leq k_1 < k_2$ and $a^{k_1} = a^{k_2}$. At step 4 in iteration k_2, we then have $a^{k_1} = a^{k_2}$ being an optimal solution to (36) with $d = d^{k_2}$. So, attack a^{k_1} is not blocked by designer decision d^{k_2}, which was generated in the previous iteration $k_2 - 1 \geq k_1$. At

Algorithm 1: Solution algorithm with optimization-based designer

```
Input: An optimization problem of the form (31), as described by some representation of
       the set $\mathcal{D}$, the set functions $\mathcal{A}(\,\cdot\,)$ and $\mathcal{X}(\,\cdot\,)$, and a tolerance $\epsilon \geq 0$
Output: A first-player solution $d^* \in \mathcal{D}$ and corresponding objective value $U$ such that
        that $U \leq \nu + \epsilon$, where $\nu$ is the optimal objective value of (31)
1  Let $k = 0$, $K = \emptyset$, $L = -\infty$, $U = \infty$
2  Select any initial first-player solution $d^0 \in \mathcal{D}$
3  while $U - L > \epsilon$ do
4      Fix $d = d^k$ in (36) and solve. Let $u(d^k)$ be the objective value and $a^k$ be an optimal
        solution
5      if $u(d^k) < U$ then
6          $U \leftarrow u(d^k)$
7          $d^* \leftarrow d^k$
8      end
9      $K \leftarrow K \cup \{k\}$
10     Solve (33)-(35), letting the optimal solution be $d^{k+1}$ and its optimal value be $L$.
11     $k \leftarrow k + 1$
12 end
13 return $(d^*, U)$
```

iteration $k_2 - 1$, attack a^{k_1} can therefore not have been blocked and the first disjunct had to have been selected in (35) for $k = k_1 \in K$. So, after the solution of (33)-(35) in iteration $k_2 - 1$, one has $L = \xi \geq \eta_{k_1} = f(x^{k_1})$, where x^{k_1} optimizes f over $\mathcal{X}(a^{k_1})$. Returning now to iteration k_2, the algorithm inherits $L \geq f(x^{k_1})$ and by hypothesis regenerates a^{k_1} as the optimal solution to (36) in step 4. This means that $u(d^{k_2}) = f(x^{k_1}) = f(x^{k_2})$, the minimum value of f over $\mathcal{X}(a^{k_1}) = \mathcal{X}(a^{k_2})$. Therefore, after steps 5-7, we will have $U \leq f(x^{k_1})$. The values of L produced in step 10 are monotonically nondecreasing, so upon reaching step 10 we will continue to have $L \geq f(x^{k_1})$ and also inherit $U \leq f(x^{k_1})$. Therefore, $U - L \leq 0 \leq \epsilon$ and the **while** loop will terminate after iteration k_2.

We conclude that the algorithm generates a new attack at each iteration in which it does not terminate. It can therefore run at most $H + 1$ iterations, since at that point there can be no new attacks to generate.

Concerning ϵ-optimality, each of the values $u(d^k)$ generated in step 4 is the objective value of an optimal attacker response to some feasible designer decision d^k, and is therefore an upper bound on ν as discussed earlier. At any iteration $\bar{k}$, the variable $U = \min_{k=0,\ldots,\bar{k}}\{u(d^k)\}$ is therefore also an upper bound on ν. As also discussed earlier, the value L produced at each iteration is the optimal value of a relaxation of (31), so we always have $L \leq \nu$. The algorithm only terminates when $U - L \leq \epsilon$, so we conclude that $U \leq L + \epsilon \leq \nu + \epsilon$ upon termination. □

Now, we show some properties of Algorithm 1 that will establish a connection to the covering decomposition algorithm of Israeli and Wood (2002).

Observation 1. *In Algorithm 1, $L = \underline{u}$ until the iteration the algorithm terminates, and the returned solution will be exactly optimal regardless of the value of ϵ.*

Proof. Consider an iteration $\bar{k}$ in which all the previous second-player solutions $\{a^1, a^2, \ldots, a^{\bar{k}-1}\}$ have been blocked at all prior iterations (that is, the second disjunct in (35) was selected). Then, after adding the solution $a^{\bar{k}}$, either it is possible to block all of $\{a^1, a^2, \ldots, a^{\bar{k}}\}$ — that is, find some $d^{\bar{k}} \in \mathcal{D}$ such that $a^1, \ldots, a^{\bar{k}} \notin \mathcal{A}(d^{\bar{k}})$ — or it is not. If so, the lower bound remains $\underline{u}$. If not, the attacks $\{a^1, a^2, \ldots, a^{\bar{k}}\}$ cannot all be blocked by one $d \in \mathcal{D}$, and the optimal objective value L of (33)-(35) at iteration $\bar{k}$ must be equal to $u(d^i)$ for some $i \in \{0, \ldots, \bar{k}\}$. Now, $U = u(d^*) \leq u(d^i)$ since it is the smallest optimal value seen so far in step 4. But since $L = u(d^i)$, we conclude that $U \leq u(d^i) = L \leq L + \epsilon$, so the **while** loop in the algorithm will terminate following iteration $\bar{k}$ with an exact solution to the problem. □

Observation 1 yields a simpler algorithm: we need never consider the disjunction in (35), but may instead solve a feasibility problem including only the constraints in the second disjunct. That is, we find a solution d in the set defined by

$$\mathcal{D}(K) := \{d \in \mathcal{D} : a^k \notin \mathcal{A}(d) \, \forall k \in K\} \tag{37}$$

until $\mathcal{D}(K)$ is empty. The resulting algorithm is essentially the covering decomposition algorithm from Israeli and Wood (2002), treating the trilevel problem as a bilevel problem with a bilevel follower. It is presented, after some refinements described in the next section, as Algorithm 2 below.

It would have been possible to skip the presentation of Algorithm 1 and instead develop Algorithm 2 directly. We chose to start with Algorithm 1 for several reasons: first, we believe that the connection between the two algorithms is of interest and has not been clearly elucidated in the literature before. Second, there are straightforward modifications to Algorithm 1 that would still yield a correctly convergent method, but to which Observation 1 would no longer apply. In this case, the lower bounds L could potentially evolve in a nontrivial manner. Therefore, the value of ϵ might have a bearing on the algorithm output and it might be possible to terminate early with a suboptimal solution of known quality. Modifications to Algorithm 1 that could lead to such behavior include initializing K to be a nonempty set of attacks that are not simultaneously blockable, or supplementing the optimal solutions to the attacker problem (36) generated in step 4 with additional, suboptimal solutions. The practicality of such approaches would depend on the tractability of the designer's resulting disjunctive program (33)-(35). As we shall see in Section 4, however, the designer problem can be very challenging even in the simplified situation in which the only goal is to block all the attacks known so far, so being able to solve (33)-(35) could well prove to be a major difficulty in using such possible extensions to Algorithm 1.

3.1 Some Variations on the Covering Decomposition Algorithm

Next, we make a new observation regarding the covering decomposition algorithm (Israeli and Wood 2002): in Algorithm 1, it is sometimes unnecessary to solve (36) to optimality. Specifically, for iterations k in which the attacker problem (36) has an optimal value greater than or equal to the current upper bound U on the optimal value, it suffices in step 4 to compute any attack a^k inflicting damage of at least U on the designer solution d^k, as opposed to an attack with the greatest possible damage. Any such attack will fail the test $u(d^k) < U$ at step 5 in the same way as the optimal one, so U will not be updated. The finiteness and correctness of the algorithm only depend on a^k being an optimal attack in the iterations k in which U is updated, so neither is affected by this change. When solving (36) with a standard MIP solver, one may use a "cutoff" option to allow the solver to terminate early if it finds a feasible solution with objective value at least U. The modified algorithm is given in Algorithm 2. By Proposition 1, this algorithm returns an optimal first-player solution to (36) after a finite number of iterations.

Algorithm 2: Solution algorithm with feasibility-based designer

```
Input: An optimization problem of the form (31), as described by some representation of
       the set $\mathcal{D}$ and the set functions $\mathcal{A}(\,\cdot\,)$ and $\mathcal{X}(\,\cdot\,)$
Output: An optimal first-player solution $d^* \in \mathcal{D}$ to (31) and its objective value
1  Let $k = 0$, $K = \emptyset$, $U = \infty$
2  Select any initial first-player solution $d^0 \in \mathcal{D}$.
3  repeat
4      Fix $d = d^k$ in (36) and solve. If the optimal value is at least $U$, any solution $a^k$ with
        objective at least $U$ may be returned; if the optimal objective value is below $U$, then
        let $a^k$ be an optimal solution and $u(d^k)$ be the corresponding objective value
5      if $u(d^k) < U$ then
6          $U \leftarrow u(d^k)$
7          $d^* \leftarrow d^k$
8      end
9      $K \leftarrow K \cup \{k\}$
10     Attempt to solve the feasibility problem (37), either finding some element $d^{k+1} \in \mathcal{D}(K)$
        or determining that $\mathcal{D}(K) = \emptyset$
11     $k \leftarrow k + 1$
12 until $\mathcal{D}(K) = \emptyset$
13 return $(d^*, U)$
```

Last, it is also possible to initialize the set K in Algorithm 2 as a set of precomputed second-player solutions, as long as the first-player problem is feasible for this set, that is, the first player can block all the solutions. While we leave this investigation for future work, it seems possible that a careful choice of initial attacks should reduce the number of iterations needed for convergence.

3.2 Applying the Algorithm for Power System SCADA Security

For our application, Algorithm 2 will fall victim to symmetry in the formulation of $\mathcal{D}(K)$ if multiple representations of equivalent first-player solutions do not block the same set of second-player solutions. For example, in the optimal network segmentation problem, we define attacks to be the set of relays that are compromised, rather than the set of enclaves that the attacker infiltrates, as in Arguello et al. (2021). If attacks were to be represented as the set of compromised enclaves, Algorithm 2 would be guaranteed to enumerate through every possible relabeling of the enclaves before being able to fully block each attack.

3.2.1 The designer feasibility problem

With this representation in mind, we now present our formulation of the feasibility problem (37). This formulation introduces two additional families of variables: $t_{n,i,k}$ indicates whether enclave i in node n has to be compromised to execute attack k, and $\tau_{s,i,k}$ indicates whether any relay controlled by a substation s which is controlled by enclave i in its parent control center must be compromised to execute attack k. Next, for notational convenience, let

$$K(n) = \{k \in K : R(n) \cap A(k) \neq \emptyset\}, \tag{38}$$

that is, $K(n)$ is the set of attacks which compromise at least one relay descended from node n. Similarly, let

$$\mathcal{B}(k) = \{n \in \mathcal{B} : A(k) \cap R(n) \neq \emptyset\} \tag{39}$$
$$\mathcal{C}(k) = \{n \in \mathcal{C} : A(k) \cap R(n) \neq \emptyset\}. \tag{40}$$

$\mathcal{B}(k)$ and $\mathcal{C}(k)$ are the sets of balancing authorities and control centers, respectively, that must be compromised to execute attack k. Then, for a set of attacks K, the designer feasibility problem may be expressed by appending the following to (2)-(13):

$$t_{n,i,k} \leq \sum_{r \in R(n) \cap A(k)} q_{r,i} \qquad \forall n \in \mathcal{C}, i \in E(n), k \in K(n) \tag{41}$$
$$t_{n,i,k} \leq \sum_{v \in C(n) \cap \mathcal{C}(k)} \sum_{j \in E(v)} t_{v,j,k} y_{v,j,i} \qquad \forall n \in \mathcal{B}, i \in E(n), k \in K(n) \tag{42}$$
$$\tau_{s,i,k} \leq \sum_{r \in C(s) \cap A(k)} q_{r,i} \qquad \forall s \in \mathcal{S}, i \in E(P(s)), k \in K(s) \tag{43}$$
$$\sum_{s \in \mathcal{S}} \sum_{j \in E(P(s))} \tau_{s,j,k} + \sum_{n \in \mathcal{B}(k) \cup \mathcal{C}(k)} \sum_{i \in E(n)} t_{n,i,k} \geq W + 1 \qquad \forall k \in K \tag{44}$$
$$\tau_{s,i,k} \in \{0,1\} \qquad \forall s \in \mathcal{S}, i \in E(P(s)), k \in K(s) \tag{45}$$
$$t_{n,i,k} \in \{0,1\} \qquad \forall n \in \mathcal{C} \cup \mathcal{B}, i \in E(n), k \in K(n) \tag{46}$$

The constraints (41) enforce that, unless a control center enclave is an ancestor of a relay which is attacked in an attack k, that enclave is not necessary to accomplish attack k. Similarly, the constraints (42) enforce that a balancing authority enclave is not necessary for attack k unless one of its children is. While we write (42) in a nonlinear form, the nonlinearities are products of binary variables which we linearized via standard reformulations in our computational experiments. The constraints (43) effectively define $\tau_{s,i,k}$ to be an indicator of whether at least one relay controlled by substation s and attacked by k has enclave i in $P(s)$ as an ancestor. The constraints (44) require that all attacks in K are over budget and therefore blocked.

3.2.2 Inner bilevel problem.

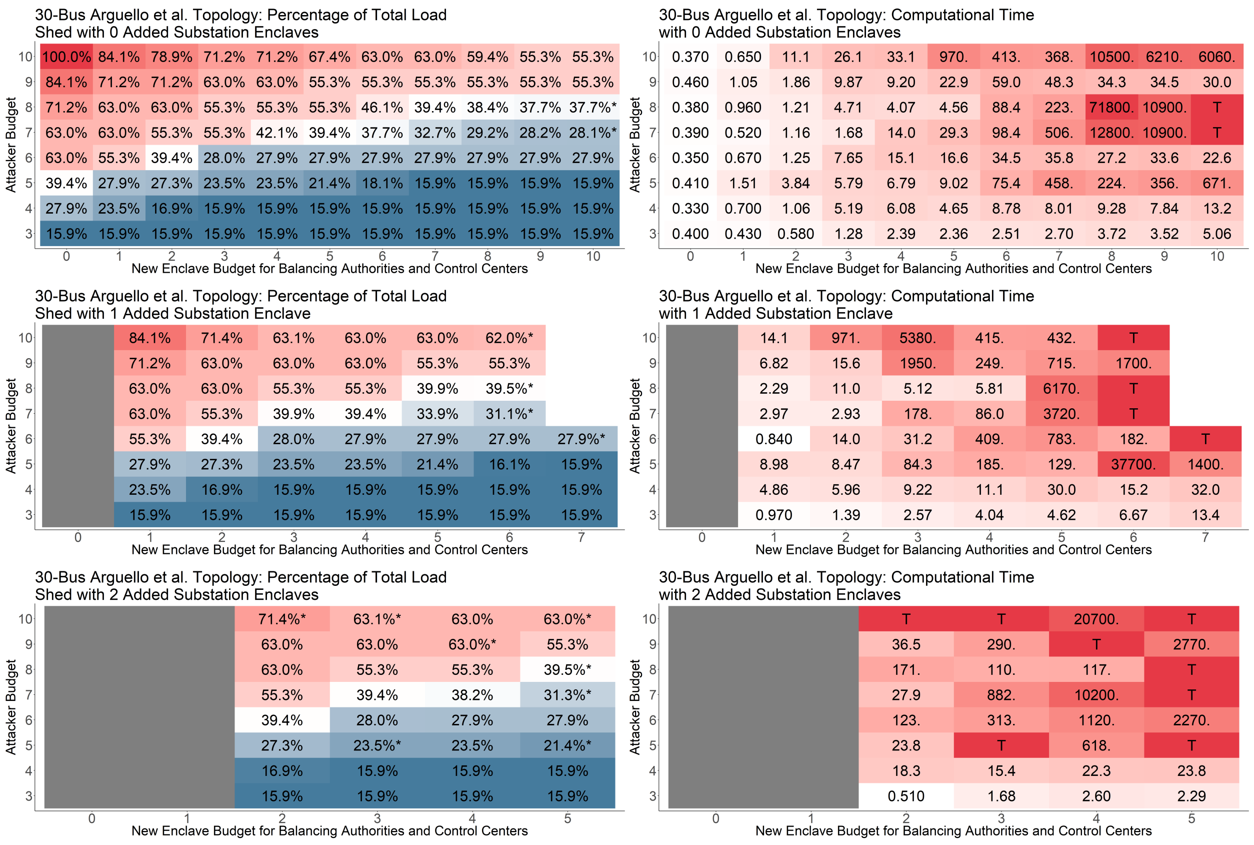
Ideally, the third-player re-dispatch problem would be as realistic a model of power grid operations as possible, suggesting an AC optimal power flow (ACOPF) formulation (Alderson et al. 2011). Unfortunately, the ACOPF equations are nonconvex (Carpentier 1962), which would likely make (36) computationally prohibitive. For this reason, it is common to approximate ACOPF linearly, using the DCOPF formulation (O'Neill et al. 2011, Kocuk et al. 2016). Though DCOPF is a linear program, various authors including Sundar et al. (2018) and Johnson and Dey (2021) find that solving (36) via dualize and combine with DCOPF is still computationally challenging for large networks and large attacker budgets. Johnson and Dey (2021) show that in contexts where the objective is to minimize load shed and having correct line flows is not important, network flow is a good approximation of DCOPF. This holds with theoretical guarantees for uncongested networks, but was shown to hold empirically even for congested networks. So, as mentioned previously, we formulate (36) as the network flow restriction from Johnson and Dey (2021), thereby producing a computationally tractable problem suitable for the dualize-and-combine approach. The resulting attacker MIP is given in the appendix, and we refer the reader to Johnson and Dey (2021) for further details on the resulting single-level formulation.

4 Computational Results

In this section, we address two questions: First, what are the impacts of the attacker and defender budgets on the effectiveness of the optimal network segmentation? Second, what are the computational limitations of Algorithm 2 as applied to the network segmentation problem? To answer these questions, we ran Algorithm 2 on several test networks, each for a variety of attacker and defender budgets. We report both the objective value as a percentage of the total load in the system and the computational time to solve the model. We show that, with well-chosen defense budgets, the damage from a cyberattack can be greatly reduced by optimal network segmentation, sometimes by as much as 50%. In addition, most of the improvement can be achieved via the combined balancing authority and control center defense budget. While increases in the substation budget sometimes further reduce the load shed of a worst-case attack, the reduction is usually moderate. However, increases in the substation budget greatly increase the computational time of Algorithm 2. Last, we show that, even for a realistically sized grid, we can use Algorithm 2 to solve network segmentation problems with large enough defender budgets to significantly reduce the impact of a worst-case attack that initially shed as much as 22% of the total system load.

We present results on three different power grid networks: the IEEE 30-bus case, a 500-bus network, and a 2,000 bus test system based on the ERCOT (Electric Reliability Council of Texas) network. For the 30-bus case, we use the communication network from Arguello et al. (2021). For the 500-bus case, we present results using two different communication networks: one has a single balancing authority and 5 control centers, and the other has 2 balancing authorities and 6 control centers. The substations are assigned to the control centers so that children of the same control center are nearby geographically. For the 2,000-bus system, we generated a control network with 2 balancing authorities and 5 control centers.

Our model is implemented in Pyomo (Hart et al. 2011, Bynum et al. 2021), and we use Gurobi version 9.0.2 to solve both the feasibility problem and the upper bounding problem (Gurobi Optimization, LLC 2020). To solve the feasibility problem, we set Gurobi's `MIPFocus` parameter to 2 (prioritizing identification of feasible solutions) and cut off the solve after the first feasible solution. The objective (required by Gurobi) is to maximize the left-hand side of (44), with k corresponding to the most recently generated attack. We chose this objective since it rewards blocking the most recently generated attack; however, since we are mainly relying on Gurobi's primal heuristics, it is unclear how much the choice of objective matters. All experiments were run giving Gurobi 12 threads on a server with 40 Intel Xeon 2.20GHz CPUs and 256GB of RAM (our experience being that allocating more than 12 threads to Gurobi in this environment does not improve performance).



30-Bus Arguello et al. Topology: Percentage of Total Load Shed with 0 Added Substation Enclaves

Attacker Budget	0	1	2	3	4	5	6	7	8	9	10
10	100.0%	84.1%	78.9%	71.2%	71.2%	67.4%	63.0%	63.0%	59.4%	55.3%	55.3%
9	84.1%	71.2%	71.2%	63.0%	63.0%	55.3%	55.3%	55.3%	55.3%	55.3%	55.3%
8	71.2%	63.0%	63.0%	55.3%	55.3%	55.3%	46.1%	39.4%	38.4%	37.7%	37.7%*
7	63.0%	63.0%	55.3%	55.3%	42.1%	39.4%	37.7%	32.7%	29.2%	28.2%	28.1%*
6	63.0%	55.3%	39.4%	28.0%	27.9%	27.9%	27.9%	27.9%	27.9%	27.9%	27.9%
5	39.4%	27.9%	27.3%	23.5%	23.5%	21.4%	18.1%	15.9%	15.9%	15.9%	15.9%
4	27.9%	23.5%	16.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%
3	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%

30-Bus Arguello et al. Topology: Computational Time with 0 Added Substation Enclaves

Attacker Budget	0	1	2	3	4	5	6	7	8	9	10
10	0.370	0.650	11.1	26.1	33.1	970.	413.	368.	10500.	6210.	6060.
9	0.460	1.05	1.86	9.87	9.20	22.9	59.0	48.3	34.3	34.5	30.0
8	0.380	0.960	1.21	4.71	4.07	4.56	88.4	223.	71800.	10900.	T
7	0.390	0.520	1.16	1.68	14.0	29.3	98.4	506.	12800.	10900.	T
6	0.350	0.670	1.25	7.65	15.1	16.6	34.5	35.8	27.2	33.6	22.6
5	0.410	1.51	3.84	5.79	6.79	9.02	75.4	458.	224.	356.	671.
4	0.330	0.700	1.06	5.19	6.08	4.65	8.78	8.01	9.28	7.84	13.2
3	0.400	0.430	0.580	1.28	2.39	2.36	2.51	2.70	3.72	3.52	5.06

30-Bus Arguello et al. Topology: Percentage of Total Load Shed with 1 Added Substation Enclave

Attacker Budget	0	1	2	3	4	5	6	7
10		84.1%	71.4%	63.1%	63.0%	63.0%	62.0%*	
9		71.2%	63.0%	63.0%	63.0%	55.3%	55.3%	
8		63.0%	63.0%	55.3%	55.3%	39.9%	39.5%*	
7		63.0%	55.3%	39.9%	39.4%	33.9%	31.1%*	
6		55.3%	39.4%	28.0%	27.9%	27.9%	27.9%	27.9%*
5		27.9%	27.3%	23.5%	23.5%	21.4%	16.1%	15.9%
4		23.5%	16.9%	15.9%	15.9%	15.9%	15.9%	15.9%
3		15.9%	15.9%	15.9%	15.9%	15.9%	15.9%	15.9%

30-Bus Arguello et al. Topology: Computational Time with 1 Added Substation Enclave

Attacker Budget	0	1	2	3	4	5	6	7
10		14.1	971.	5380.	415.	432.	T	
9		6.82	15.6	1950.	249.	715.	1700.	
8		2.29	11.0	5.12	5.81	6170.	T	
7		2.97	2.93	178.	86.0	3720.	T	
6		0.840	14.0	31.2	409.	783.	182.	T
5		8.98	8.47	84.3	185.	129.	37700.	1400.
4		4.86	5.96	9.22	11.1	30.0	15.2	32.0
3		0.970	1.39	2.57	4.04	4.62	6.67	13.4

30-Bus Arguello et al. Topology: Percentage of Total Load Shed with 2 Added Substation Enclaves

Attacker Budget	0	1	2	3	4	5
10			71.4%*	63.1%*	63.0%	63.0%*
9			63.0%	63.0%	63.0%*	55.3%
8			63.0%	55.3%	55.3%	39.5%*
7			55.3%	39.4%	38.2%	31.3%*
6			39.4%	28.0%	27.9%	27.9%
5			27.3%	23.5%*	23.5%	21.4%*
4			16.9%	15.9%	15.9%	15.9%
3			15.9%	15.9%	15.9%	15.9%

30-Bus Arguello et al. Topology: Computational Time with 2 Added Substation Enclaves

Attacker Budget	0	1	2	3	4	5
10			T	T	20700.	T
9			36.5	290.	T	2770.
8			171.	110.	117.	T
7			27.9	882.	10200.	T
6			123.	313.	1120.	2270.
5			23.8	T	618.	T
4			18.3	15.4	22.3	23.8
3			0.510	1.68	2.60	2.29

Figure 5: Computational results from running Algorithm 2 on the 30-bus test instance.

4.1 30-Bus Test Case

First, we discuss results for the 30-bus test system. Because of the small number of buses, we are able to solve instances of the network segmentation problem for attacker budgets of between 3 and 10 enclaves (attacking even a single relay requires compromising its substation, the substation's parent control center, and the control center's balancing authority, so the minimum budget to execute any attack is 3). Without any segmentation, an attacker budget of 10 is sufficient to shed all the load in the system. The results are depicted in Figure 5. We consider balancing authority and control center enclave budgets of between 0 and 10, and substation enclave budgets of 0, 1, and 2. In the first column of plots, we show the optimal objective value (or, for the ones marked with an asterisk, the best upper bound U in the case that Algorithm 2 does not converge within 24 hours) as a percentage of the total load in the system. The cells are colored from blue to white to red, where blue, white and red respectively correspond to the minimum, mean, and maximum percentage load shed across all the plots. In the second column, we show the computational time in seconds. The cells are colored according to the log of the computational time, from white to red, where white is the minimum run time in the table and red is 24 hours. We use 'T' to denote that Algorithm 2 did not converge within 24 hours. The first row of plots has a budget of 0 substation enclaves for the defender, the second row has a budget of 1 substation enclave, and the last row has a budget of 2 substation enclaves. Figures 6-8 use the same format.

We include fewer runs for the higher substation enclave budgets since they became computationally expensive. Note, however, that as the substation enclave budget increases, it is rare to get a large reduction in load shed. For example, between budgets of 0 and 1 substation enclaves, only 13 of the budget combinations benefit from the additional substation enclave. However, on average, the computational time increases by more than 2 hours. Comparing budgets of 1 and 2 substation enclaves, only 5 of the instances we solve see any improvement in load shed and the computational time increases by more than 6 hours on average.

Thus, for this test case, we find that most of the benefit from network segmentation is attained by investing solely in new enclaves in the balancing authority and control center levels. We can solve all but two of these instances within 24 hours. Load shed is typically reduced by about 50% as compared to the unsegmented network. The charts also indicate the trade-off between defender and attacker budgets: for any given attacker budget, there appears to be a point of diminishing returns after which more designer enclaves have little or no impact on total load shed; the location of this point increases with the attacker budget.

4.2 500-Bus Test Cases

Next, we present experiments on two different control network structures for the 500-bus test case. Figure 6 shows results for the communication network with one balancing authority and 5 control centers. For the larger network, we are more computationally limited in what we can solve, but it is still possible to solve the network segmentation problem on attack budgets of up to 9 enclaves, which would shed 30% of the load without segmentation. Even a budget of 3 balancing authority and control center enclaves can reduce this load shed by about a third to about 22% of total system load. Again, we see limited utility in increasing the network designer's substation enclave budget. Between budgets of 0 and 1 substation enclaves, only 10 of the instances see a reduction in load shed with the additional substation enclave, and the reduction is at most 4.3% of the total load. Instances with larger attack budgets and larger budgets for balancing authorities and control

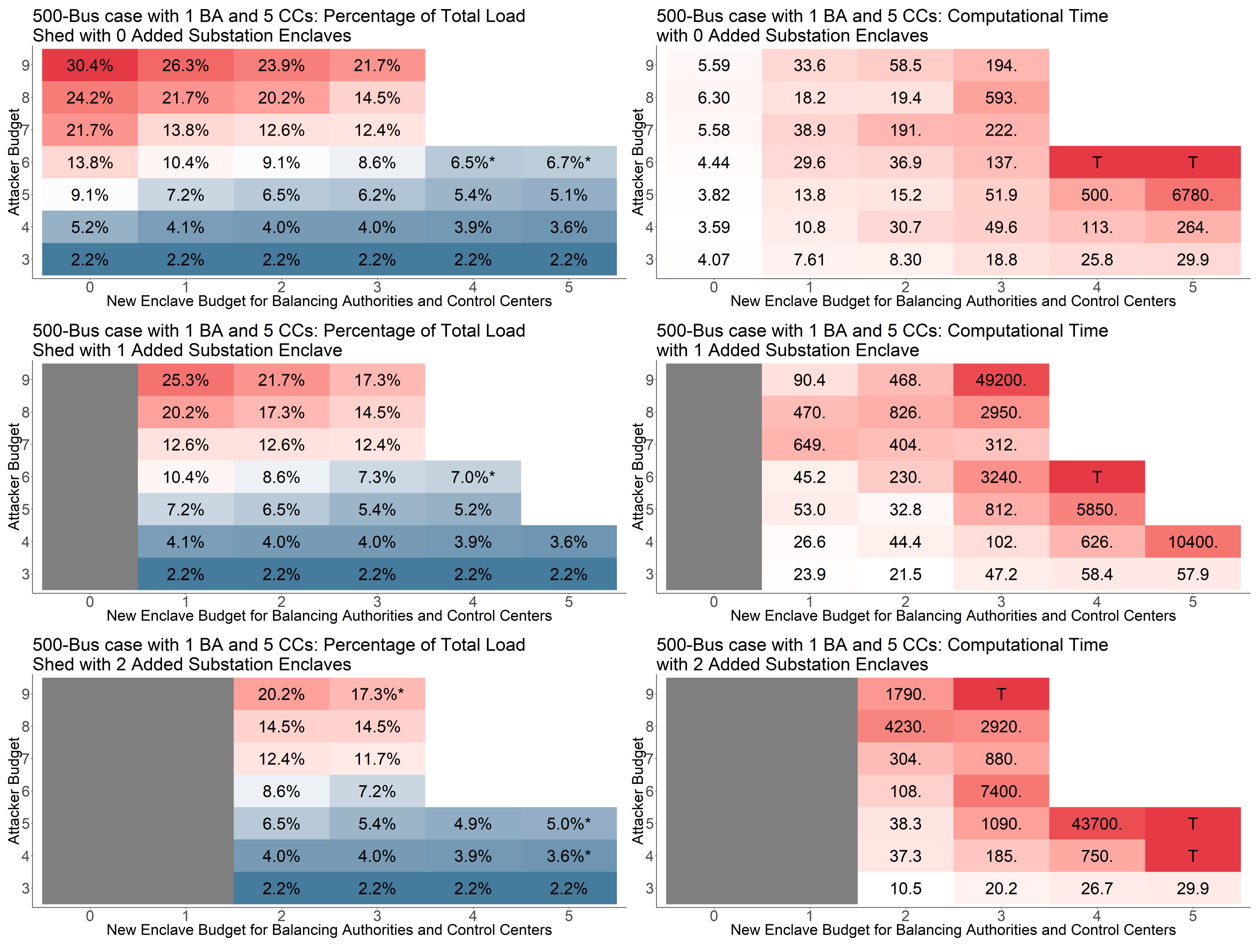


Figure 6: Computational results from running Algorithm 2 on the 500-bus test instance with a communication network with 1 balancing authority (BA) and 5 control centers (CCs).

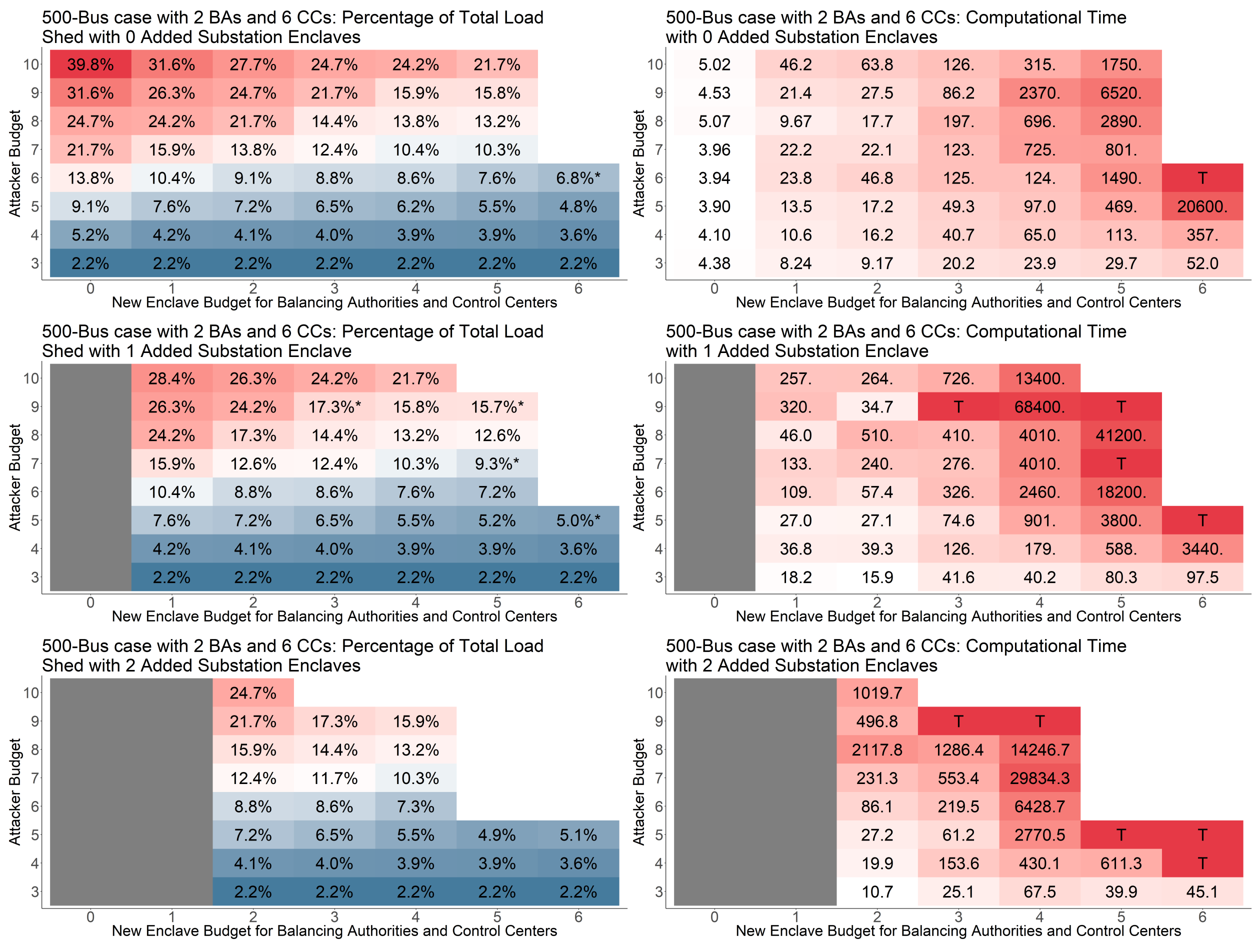


Figure 7: Computational results from running Algorithm 2 on the 500-bus test instance with a communication network with 2 balancing authorities (BAs) and 6 control centers (CCs).

centers benefit the most from having the additional substation enclave. Much as before, however, this comes at the cost of an additional 45 minutes of computational time, on average. Similarly, after increasing the substation enclave budget from 1 additional enclave to 2, only 7 instances have a reduction in load shed, by at most 2.8% of the total load. The average increase in computational time is more than 2 hours.

Next, Figure 7 depicts results for the same physical grid but with a different communication network topology having 2 balancing authorities and 6 control centers. The features observed in the previous two test cases are again present: the maximum benefit from increasing the substation enclave budget from 0 to 1 enclave is again 4.3% of the total load, and only 20 of the 42 instances have any benefit. The computational time increases by more than 3 hours on average. Increasing from 1 substation enclave to 2 substation enclaves, 7 instances benefit in terms of load shed, by at most 2.6%. The computational time increases by more than 2 hours on average between these two sets of experiments.

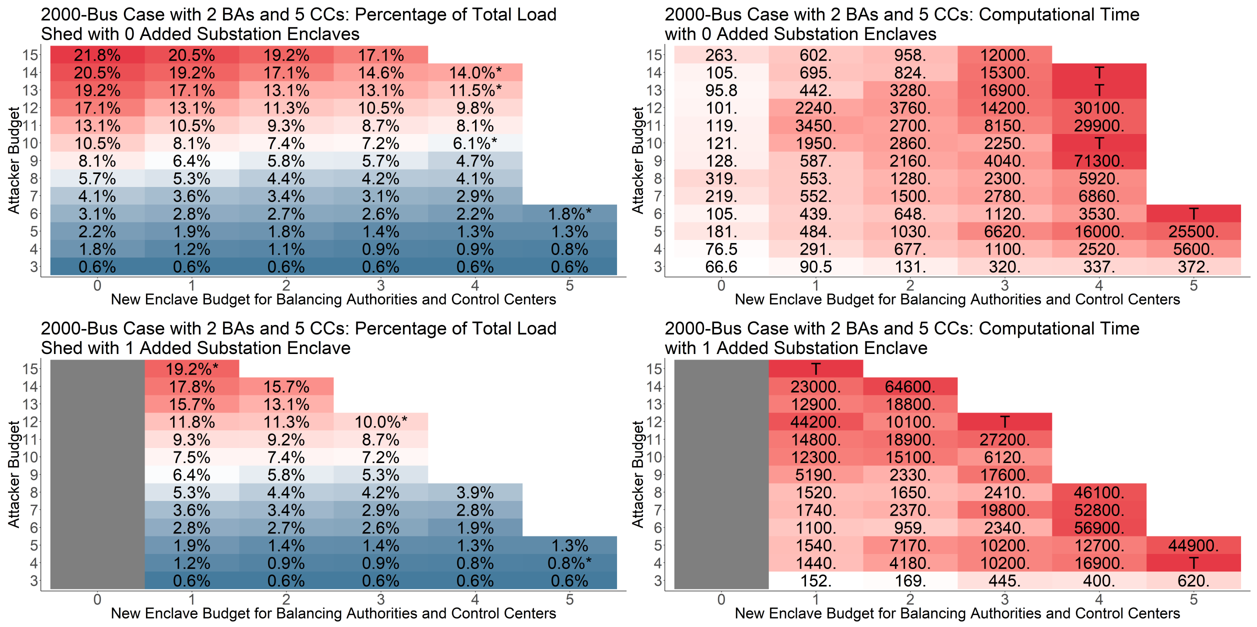


Figure 8: Computational results from running Algorithm 2 on the 2,000-bus test instance with a communication network with 2 balancing authorities (BAs) and 5 control centers (CCs).

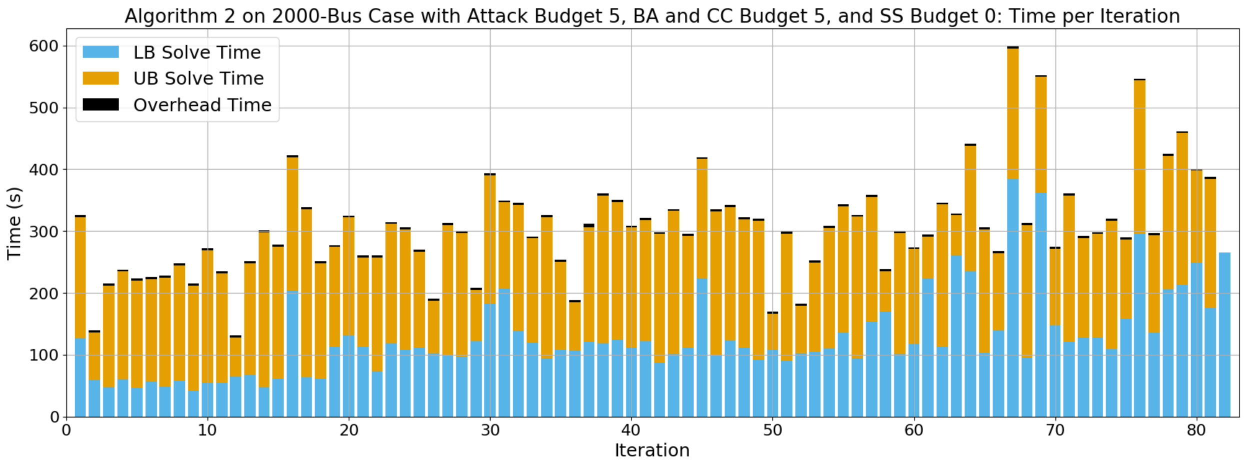
4.3 2,000-Bus Test Case

Finally, Figure 8 depicts results on the 2,000-bus case based on the ERCOT grid, considering attacker budgets which can shed up to 22% of the total load in the absence of network segmentation. Computationally, these instances are much more challenging, but many are still solvable within 24 hours. Even more significantly than in the cases discussed previously, additional substation enclaves are of little benefit to the network designer. With an increase in the budget from 0 to 1, only 20 of the instances have decreased load shed, by at most 1.4% of the total load. To solve the instances with the larger substation enclave budget, however, takes 4.5 hours longer on average. For this test case, we did not perform any experiments with a substation enclave budget of 2, since a budget of 1 was already very computational intensive and demonstrated only modest benefit. Overall, we observe that Algorithm 2 is capable of solving the network segmentation problem on a realistically sized grid, and for defender budgets that can decrease the load shed from a relatively severe worst-case attack by at least 20%, and sometimes considerably more: for example a defender budget of 4 can reduce the worst-case load shed from an attacker budget of 9 from 8.1% to 4.7%.

4.4 Performance Breakdown for Algorithm 2

While the previous section presented the total computational time for Algorithm 2, this section compares the time spent in the feasibility problem (the designer) and the upper-bounding problem (the attacker). As one might expect, the designer feasibility problem dominates the computational time for more challenging problems since it grows with each iteration. However, while it does not vary much over successive iterations, the attacker problem solution time can be significant for the larger test network.

Figure 9 shows plots of the computational time per iteration of Algorithm 2 for two different



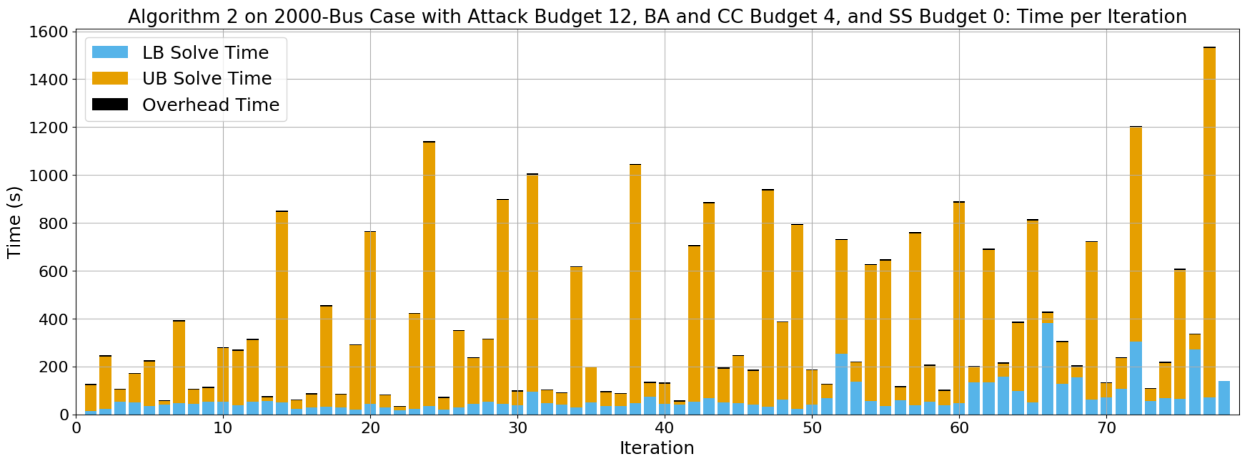


Figure 9: Computational time per iteration solving two different attacker and defender budget combinations on the 2,000-bus case. The first has a budget of 5 enclaves for the balancing authorities (BA) and control centers (CC), 0 enclaves for the substations (SS) and an attack budget of 5 enclaves. The second has a budget of 4 enclaves for the balancing authorities and control centers, no budget at the substation level, and an attack budget of 12 enclaves. 'LB Solve Time' and 'UB Solve Time' are the Gurobi solve times for the feasibility problem and the upper-bounding problem respectively. 'Total Time' is the sum of 'LB Solve time' and 'UB Solve Time', plus some additional overhead from our implementation.

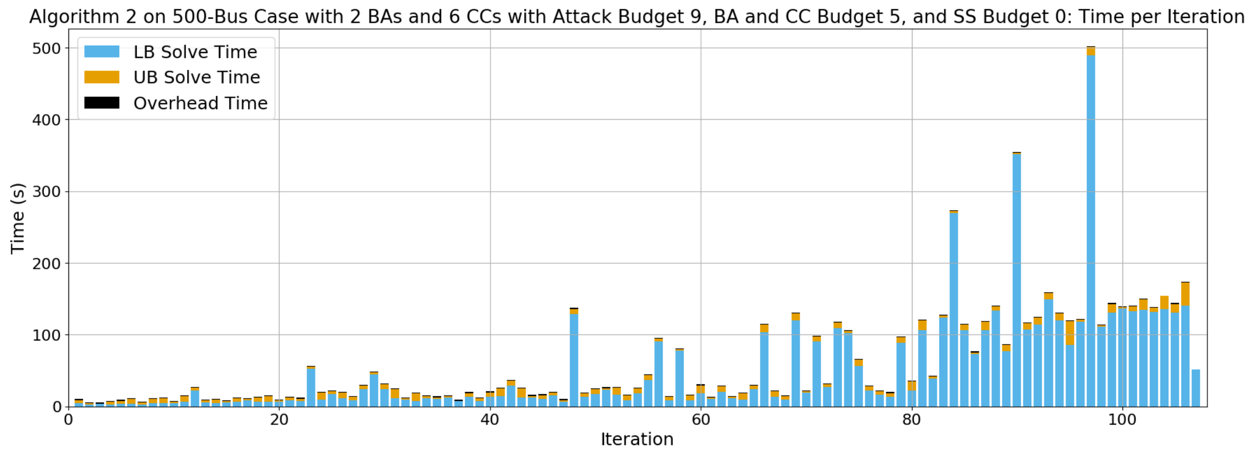


Figure 10: Computational time per iteration solving for a budget of 5 balancing authorities (BA) and control centers (CC), no additional substation (SS) enclaves, and an attack budget of 9 enclaves on the 500-bus case with 2 balancing authorities and 6 control centers. The color-coding scheme is the same as in Figure 9.

budget combinations on the 2,000-bus case. We plot the Gurobi solve times for the feasibility problem and the upper-bounding problem, as well as the time taken for the entire iteration, including the overhead in our implementation. In the first plot, the attacker only has a budget of 5 enclaves, making the upper-bounding (attacker) problem relatively easy. Its time remains approximately constant as iterations increase, whereas the feasibility problem, plotted in the 'LB Solve Time' series tends to increase with the iterations. In contrast, in the second plot, the attacker has a higher budget, making the upper-bounding problem more difficult. In this run, the upper-bounding problem dominates the total time, and varies greatly between iterations. While the time for the feasibility problem increases slightly with the iterations, the time for the upper-bounding problem, while quite variable, does not appear much influenced by the iteration count.

In contrast to the results in Figure 9, Figure 10 shows a plot of computational times per iteration for a particular budget on the 500-bus case with the 2-balancing-authority and 6-control-center SCADA network. In this plot, despite the attacker budget being relatively high for the instance, the total time is still dominated by the feasibility problem solution time. This outcome is unsurprising in light of the results from Johnson and Dey (2021), since for a 500-bus network, we do not expect the upper-bounding problem to be difficult. Its time is nearly constant with iterations, whereas the time for the feasibility problem increases considerably.

Overall, we find that for smaller physical networks and for smaller attack budgets, the feasibility problem dominates the computational time in the algorithm. However, for large attack budgets on large physical networks, the upper-bounding problem can also be challenging, sometimes so much that it becomes dominant.

4.5 Benefit of Subproblem Cutoffs

This section explores the utility of using cutoffs when solving the upper-bounding problem (36). To do so, we make a comparison between Algorithm 2 as written and a version wherein we always solve the attacker problem to optimality. In Table 1, we compare these variations for various budgets

Table 1: Impact of the attacker problem objective cutoff described in Section 3.1. The boldface entries denote the version with fewer iterations in the "Number of Iterations" column and less time in the last two columns. Recall that V, T, and W are the substation enclave budget, balancing authority and control center enclave budget, and attacker budget respectively.

				Number of Iterations		Total Time (s)		Mean Problem (36) Time (s)	
Test Case	V	T	W	With Cutoff	Without Cutoff	With Cutoff	Without Cutoff	With Cutoff	Without Cutoff
30-bus	0	5	10	33	33	970.4	**844.9**	0.1	0.1
	0	7	7	**28**	29	506.3	**451.5**	0.2	0.2
	0	9	5	**35**	38	**355.9**	403.1	0.2	0.2
	0	9	10	34	**29**	6206.2	**1427.6**	0.1	0.1
	1	3	6	58	**30**	31.2	**11.1**	0.1	0.1
	1	4	7	**42**	53	**86.0**	185.9	0.1	0.1
	2	2	8	**113**	149	**171.5**	324.1	0.1	0.1
500-bus, 1BA5CC	0	2	9	**11**	16	**58.5**	135.1	**2.9**	6.1
	0	3	7	29	**21**	221.9	**186.2**	**4.6**	5.0
	0	4	5	**52**	54	**500.2**	575.0	**2.7**	3.1
	1	2	6	**33**	35	230.0	**162.0**	5.9	**3.7**
	2	3	5	**153**	205	**1087.7**	1863.5	**2.9**	3.4
	1	4	5	**160**	170	5846.6	**3394.2**	6.1	**3.4**
	2	2	6	26	**22**	**108.2**	125.3	**3.2**	5.2
	2	4	4	87	**79**	750.4	**671.8**	**1.8**	1.9
	1	2	8	67	**49**	826.3	**397.9**	11.2	**7.2**
500-bus, 2BA6CC	0	3	8	**34**	38	**197.1**	278.2	**3.5**	4.0
	0	4	10	36	**31**	315.1	**276.6**	**3.6**	3.8
	1	2	7	54	**47**	239.8	**199.5**	**3.0**	3.1
	1	4	5	85	**70**	900.8	**645.2**	3.0	**2.8**
	2	3	7	115	**102**	**553.4**	560.5	**2.8**	3.2
	0	4	9	**90**	150	**2372.3**	6381.9	**3.9**	4.8
	1	4	8	**156**	167	4014.5	**3456.6**	3.8	**3.4**
	2	3	8	148	**111**	1286.4	**708.0**	3.5	3.5
	2	2	8	350	**342**	**2117.8**	2150.6	**3.5**	3.8
	1	2	8	**95**	98	509.6	**478.0**	3.7	**3.6**
2000-bus	0	1	12	17	**15**	2236.7	**1933.3**	**136.8**	138.0
	0	2	8	4	4	**1278.3**	1399.3	**362.7**	397.2
	0	2	11	**13**	21	**2700.3**	6287.1	**227.8**	305.3
	0	3	6	5	5	**1115.5**	1272.8	**199.4**	226.5
	0	4	6	14	**13**	3525.1	**3317.5**	**209.2**	211.5
	1	1	8	8	8	**1523.2**	2083.6	**189.0**	259.9
	1	1	9	33	**24**	5186.3	**3678.2**	156.5	**153.4**
	1	2	6	6	6	**958.6**	1288.1	**149.5**	205.9
	1	2	9	**12**	14	**2333.4**	3466.6	**204.5**	251.3
	1	10	4	3	3	**695.2**	939.6	**206.7**	261.2

on each of the test cases. For each instance, using cutoffs or not, the table shows the number of iterations Algorithm 2 takes to converge, the total computational time for Algorithm 2, and the average time per iteration spent solving problem (36).

For all but the 2,000-bus case, using cutoffs has mixed results: the number of iterations can increase or decrease, as can the total computational time. The first use of a suboptimal attack can make the algorithm follow a different path, which can impact the number of iterations, the total time, and the difficulty of subsequent attacker and designer problems. For 30 or 500 buses, the designer feasibility problem dominates the computational time, so the number of iterations is the principal factor determining total run time, and total iterations respond somewhat unpredictably to the use of cutoffs.

In contrast, for the 2,000 bus case, the upper-bounding problem is more difficult. As we see in the 'Mean Problem (36) Time' columns, using cutoffs does tend to decrease the time spent in the upper-bounding problem for this case, decreasing the total computational time for Algorithm 2 with only a few exceptions. Therefore, it seems that for large physical networks where solving the worst-case attack problem is computationally challenging, the opportunity to terminate the attacker solve early can pay off in the overall computational time.

5 Conclusions and Future Work

We conclude that our algorithms make trilevel optimization network segmentation models appear computationally tractable for realistic-sized power networks, at least for modest attacker and defender budgets. This may be a significant contribution, since network segmentation approaches capture the properties of real cyberattacks better than traditional models in which the designer can make particular physical assets invulnerable to any attack.

In reality, designer and attacker budgets will not simply be given and fixed. One may instead view the designer as having a decision problem with three competing objectives: minimize the number of enclaves that need to be introduced, maximize the number of enclaves an attacker needs to penetrate, and contain load shed. Charts such as those in Figures 5-8 could help a designer decide how many enclaves to create and what kind of benefit to expect from them, and the solutions to the underlying problems could guide the designer in locating the enclaves.

In future work, enhancements to the solution techniques for solving both the feasibility problem and the attacker problem could significantly reduce the time for Algorithm 2 to converge and thus broaden the range of designer and attacker budgets that are solvable. In particular, the constraints (14) only partially remove the possible symmetries in describing equivalent designer decisions, so eliminating further symmetries might be very beneficial in solving instances with larger budgets. Another possibility that might bear some investigation is solving the designer feasibility problems by some other means than a standard MIP solver, for example, using a constraint logic programming solver. Also, techniques such as Benders decomposition might be applicable to the worst-case attack problem on large physical grids (Wood 2011).

If the disjunctive lower-bounding problem (33)-(35) can be made tractable, variations on Algorithm 1 in which the designer problem is a disjunctive optimization problem instead of a feasibility problem could prove useful on the network segmentation problem and other trilevel interdiction problems. One advantage the disjunctive optimization approach might be the ability to find non-trivial lower bounds and hence meaningful optimality gaps that could allow for early termination with an approximate solution of known quality.

MIP solvers with more effective parallelism, which we expect will eventually be developed, might also be very beneficial to our algorithms.

Acknowledgments

We would like to thank Bryan Arguello and Jared Gearhart for helpful conversations and help with computational experiments during this work.

This work was supported by Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. 2021-15690 O. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

Alderson D, Brown G, Carlyle M, Wood RK (2011) Solving defender-attacker-defender models for infrastructure defense. Wood RK, Dell RF, eds., *Operations Research, Computing, and Homeland Defense, Proceedings of the 12th INFORMS Computing Society Conference*, 28–49 (Hanover, MD: INFORMS).

Alguacil N, Delgadillo A, Arroyo JM (2014) A trilevel programming approach for electric grid defense planning. *Computers & Operations Research* 41:282–290.

Arguello B, Johnson ES, Gearhart JL (2021) A trilevel model for segmentation of the power transmission grid cyber network. Technical Report 2108.10958, ArXiv.

Australian Cyber Security Centre (2020) Implementing network segmentation and segregation. URL `https://www.cyber.gov.au/sites/default/files/2020-01/PROTECT%20-%20Implementing%20Network%20Segmentation%20and%20Segregation%20%28April%202019%29.pdf`, Australian Signals Directorate.

Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, Siirola JD, Watson JP, Woodruff DL (2021) *Pyomo — optimization modeling in Python*, volume 67 of *Springer Optimization and its Applications* (New York: Springer), third edition.

Carpentier J (1962) Contributions to the economic dispatch problem. *Bulletin Society Francaise Electriciens* 8(3):431–447.

Ding T, Yao L, Li F (2018) A multi-uncertainty-set based two-stage robust optimization to defender–attacker–defender model for power system protection. *Reliability Engineering & System Safety* 169:179–186.

Fang Y, Zio E (2017) Optimizing the resilience of interdependent infrastructure systems against intentional attacks. *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, 62–67.

Fischetti M, Ljubić I, Monaci M, Sinnl M (2017) A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs. *Operations Research* 65(6):1615–1637.

Genge B, Haller P, Kiss I (2017) Cyber-security-aware network design of industrial control systems. *IEEE Systems Journal* 11(3):1373–1384.

Ghorbani-Renani N, González AD, Barker K (2021) A decomposition approach for solving tri-level defender-attacker-defender problems. *Computers & Industrial Engineering* 153:107085.

Ghorbani-Renani N, González AD, Barker K, Morshedlou N (2020) Protection-interdiction-restoration: Tri-level optimization for enhancing interdependent network resilience. *Reliability Engineering & System Safety* 199:106907.

Gurobi Optimization, LLC (2020) Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Hart WE, Watson JP, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation* 3(3):219–260.

Hua B, Baldick R, Wood RK (2019) Interdiction of a mixed-integer linear system. Preprint 2019-01-7013, Optimization Online.

Israeli E, Wood RK (2002) Shortest-path network interdiction. *Networks* 40(2):97–111.

Jeroslow RG (1985) The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming* 32:146–164.

Johnson ES, Dey SS (2021) A scalable lower bound for the worst-case relay attack problem on the transmission grid. Technical Report 2105.020801, ArXiv.

Kocuk B, Dey SS, Sun A (2016) Strong SOCP relaxations for the optimal power flow problem. *Operations Research* 64(6):1177–1196.

Lai K, Illindala M, Subramaniam K (2019) A tri-level optimization model to mitigate coordinated attacks on electric power systems in a cyber-physical environment. *Applied Energy* 235:204–218.

Lozano L, Smith JC (2017a) A backward sampling framework for interdiction problems with fortification. *INFORMS Journal on Computing* 29(1):123–139.

Lozano L, Smith JC (2017b) A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research* 65(3):768–786.

O'Neill R, Dautel T, Krall E (2011) Recent ISO software enhancements and future software and modeling plans. Technical report, Federal Energy Regulatory Commission (FERC), URL `http://www.ferc.gov/industries/electric/indus-act/rto/rto-iso-soft-2011.pdf`.

Oster M, Chatterjee S, Pan F, Bakker C, Bhattacharya A, Perkins C (2020) Power system resilience through defender-attacker-defender models with uncertainty: an overview. *2020 Resilience Week (RWS)*, 11–17.

Ouyang M (2017) A mathematical framework to optimize resilience of interdependent critical infrastructure systems under spatially localized attacks. *European Journal of Operational Research* 262(3):1072–1084.

Ouyang M, Fang Y (2017) A mathematical framework to optimize critical infrastructure resilience against intentional attacks. *Computer-Aided Civil and Infrastructure Engineering* 32(11):909–929.

Ouyang M, Tao F, Huang S, Xu M, Zhang C (2018) Vulnerability mitigation of multiple spatially localized attacks on critical infrastructure systems. *Computer-Aided Civil and Infrastructure Engineering* 33(7):585–601.

Ouyang M, Xu M, Zhang C, Huang S (2017) Mitigating electric power system vulnerability to worst-case spatially localized attacks. *Reliability Engineering & System Safety* 165:144–154.

Pillitteri VY, Brewer TL (2014) Guidelines for smart grid cybersecurity. Interagency/Internal Report (NISTIR)-7628, Revision 1, National Institute of Standards (NIST).

Scaparra MP, Church RL (2008) A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research* 35(6):1905–1923.

Smith JC, Song Y (2020) A survey of network interdiction models and algorithms. *European Journal of Operational Research* 283(3):797–811.

Stouffer K, Pillitteri V, Suzanne Lightman M, Hahn AA (2016) Guide to industrial control systems (ICS) security. Special Publication 800-82, Revision 2, National Institute of Standards (NIST).

Sundar K, Coffrin C, Nagarajan H, Bent R (2018) Probabilistic *N*-*k* failure-identification for power systems. *Networks* 71(3):302–321.

Tahernejad S, Ralphs TK, DeNegre ST (2020) A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Computation* 12(4):529–568.

Tang Y, Richard JPP, Smith JC (2016) A class of algorithms for mixed-integer bilevel min–max optimization. *Journal of Global Optimization* 66(2):225–262.

US Department of Homeland Security, Industrial Control Systems Cyber Emergency Response Team (2016) Recommended practice: Improving industrial control system cybersecurity with defense-in-depth strategies. URL `https://us-cert.cisa.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf`.

Wood RK (2011) Bilevel network interdiction models: Formulations and solutions. Cochran JJ, Cox Jr LA, Keskinocak P, Kharoufeh JP, Smith JC, eds., *Wiley Encyclopedia of Operations Research and Management Science* (New York: John Wiley & Sons).

Wu X, Conejo AJ (2017) An efficient tri-level optimization model for electric grid defense planning. *IEEE Transactions on Power Systems* 32(4):2984–2994.

Yao Y, Edmunds T, Papageorgiou D, Alvarez R (2007) Trilevel optimization in power network defense. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37(4):712–718.

Yuan W, Zhao L, Zeng B (2014) Optimal power grid protection through a defender–attacker–defender model. *Reliability Engineering & System Safety* 121:83–89.

Yue D, Gao J, Zeng B, You F (2019) A projection-based reformulation and decomposition algorithm for global optimization of a class of mixed integer bilevel linear programs. *Journal of Global Optimization* 73:27–57.

Zeng B, Zhao L (2013) Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters* 41(5):457–461.

The Attacker MIP

The attacker MIP is given by the following formulation, where λ^+ and λ^- are the dual variables of constraints (28), γ are the dual variables of constraints (29), φ and ψ are the dual variables of the two constraints in (27), and μ are the dual variables of (26). In addition, we define $d(l)$ and $o(l)$ to be the origin and destination substations of line l, and $s(g)$ to be the substation where generator g is located.

$$\begin{aligned}
\max_{\substack{\delta,u,v,w,\varphi,\\ \psi,\lambda^+,\lambda^-,\gamma,\mu}} \quad & -\sum_{l\in\mathcal{L}} \overline{F}_l \cdot (v_l\lambda_l^+ + v_l\lambda_l^-) \\
& - \sum_{g\in\mathcal{G}} \overline{P}_g u_g \gamma_g + \sum_{s\in\mathcal{S}} D_s \cdot \big((1-w_s)\varphi_s + \mu_s - \psi_s\big) && \text{(47a)} \\
\text{s.t.} \quad & \text{(15)-(25)} \\
& \lambda_l^+ - \lambda_l^- + \mu_{d(l)} - \mu_{o(l)} = 0 \qquad \forall l \in \mathcal{L} \quad (f) && \text{(47b)} \\
& \mu_{s(g)} - \gamma_g \le 0 \qquad \forall g \in \mathcal{G} \quad (p) && \text{(47c)} \\
& \varphi_s + \mu_s - \psi_s \le 1 \qquad \forall s \in \mathcal{S} \quad (\ell) && \text{(47d)} \\
& 0 \le \varphi_s \le 1 \qquad \forall s \in \mathcal{S} && \text{(47e)} \\
& 0 \le \psi_s \le 1 \qquad \forall s \in \mathcal{S} && \text{(47f)} \\
& 0 \le \lambda_l^+ \le 1 \qquad \forall l \in \mathcal{L} && \text{(47g)} \\
& 0 \le \lambda_l^- \le 1 \qquad \forall l \in \mathcal{L} && \text{(47h)} \\
& 0 \le \gamma_g \le 1 \qquad \forall g \in \mathcal{G} && \text{(47i)} \\
& -1 \le \mu_s \le 1 \qquad \forall s \in \mathcal{S} && \text{(47j)}
\end{aligned}$$

We refer the reader to Johnson and Dey (2021) for the proof of the correctness of the formulation. Note that while the objective contains bilinearities, they are products of binaries and bounded continuous variables, and are therefore easily linearized with the addition of auxiliary variables.