A Generic Optimization Framework for Resilient Systems

Marc E. Pfetsch and Andreas Schmitt*

May 27, 2021

Abstract

This paper addresses the optimal design of resilient systems, in which components can fail. The system can react to failures and its behavior is described by general mixed integer nonlinear programs, which allows for applications to many (technical) systems. This then leads to a three-level optimization problem. The upper level designs the system minimizing a cost function, the middle level represents worst-case failures of components, i.e., interdicts the system, and the lowest level operates the remaining system. We describe new inequalities that characterize the set of resilient solutions and allow to reformulate the problem. The reformulation can then be solved using a nested branch-and-cut approach. We discuss several improvements, for instance, by taking symmetry into account and strengthening cuts. We demonstrate the effectiveness of our implementation on the optimal design of water networks, robust trusses, and gas networks, in comparison to an approach in which the failure scenarios are directly included into the model.

Keywords: multi-level optimization, mixed-integer nonlinear programming, interdiction problems, gas and water network design, truss topology optimization

1 Introduction

The optimal design of systems that are robust against component failures is an interesting research area leading to challenging multi-level optimization problems. Prominent examples are given by the design of technical systems, e.g., truss structures or electrical, gas, and water networks. In such robust/resilient systems, possible structural failures can be reacted to by adjusting controls like switches/actuators and generators/compressors; some examples will be discussed in more detail later. Thus, a typical multi-level structure arises as follows: the first level corresponds to the design decisions, the second level refers to (worst case) failures, and the third level admits a reaction to the decisions/failures of the first two levels.

The goal of this paper is to develop a *generic* algorithm that is able to solve the corresponding three level design problem under failures with a focus on a *mixed-integer nonlinear behavior* of the system. To specify the problem, consider a general system design optimization problem

$$\min \{F(x,y,u) : G(x,y,u) \leq \mathbb{0},\ x \in \{0,1\}^n,\ y \in \{0,1\}^m,\ u \in \mathbb{Z}^s \times \mathbb{R}^t\}. \tag{1}$$

Here, $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ represent design decisions; the components chosen by x possibly fail, while those of y do not. Moreover, $u \in \mathbb{Z}^s \times \mathbb{R}^t$ are auxiliary variables. The objective $F : \mathbb{R}^{n+m+s+t} \to \mathbb{R}$ usually incorporates fixed costs for the design (x,y) as well as operating costs depending on u. The constraints are represented by some function $G : \mathbb{R}^{n+m+s+t} \to \mathbb{R}^r$. In this paper, F and G can be arbitrary nonlinear, possibly nonconvex functions. Problem (1) is already often quite challenging to solve, but we assume that it can

*Discrete Optimization Group, Department of Mathematics, TU Darmstadt, Germany

be solved in practice using mixed-integer nonlinear programming (MINLP) methods; see, e.g., Belotti et al. [8] for an introduction.

We then consider the following three-level problem

$$\min_{(x,y,u)\in\mathcal{X}} \max_{z\in\mathcal{Z}_k} \min_{(\tilde{x},\tilde{y},\tilde{u})\in\tilde{\mathcal{X}}} \big\{F(x,y,u) \,:\, \tilde{x}\leq x,\ \tilde{x}\leq \mathbb{1}-z,\ \tilde{y}\leq y\big\}. \tag{2}$$

Here,

$$\begin{aligned} \mathcal{X} &\coloneqq \{(x,y,u)\in\{0,1\}^n\times\{0,1\}^m\times(\mathbb{Z}^s\times\mathbb{R}^t) \,:\, G(x,y,u)\leq\mathbb{0}\},\\ \mathcal{Z}_k &\coloneqq \{z\in\{0,1\}^n \,:\, \mathbb{1}^\top z\leq k\},\\ \tilde{\mathcal{X}} &\coloneqq \{(\tilde{x},\tilde{y},\tilde{u})\in\{0,1\}^n\times\{0,1\}^m\times(\mathbb{Z}^{\tilde{s}}\times\mathbb{R}^{\tilde{t}}) \,:\, \tilde{G}(\tilde{x},\tilde{y},\tilde{u})\leq\mathbb{0}\}, \end{aligned}$$

for some maximal number $k\in\mathbb{N}$ of possibly failing components and a function $\tilde{G}:\mathbb{R}^{n+m+\tilde{s}+\tilde{t}}\to\mathbb{R}^{\tilde{r}}$ that represents the system behavior after failures.

The model works as follows: On the first level, (x,y) represent a system design and u the state in the setting without failures; the feasible region is given by $\mathcal{X}$ as in (1). Similarly, the third level involves a system design/behavior $(\tilde{x},\tilde{y},\tilde{u})$ after failures occurred; the corresponding recourse set is $\tilde{\mathcal{X}}$. One can use $\tilde{\mathcal{X}}=\mathcal{X}$, but often it makes sense to consider a possibly reduced function of the system, e.g., only a reduced amount of electricity/gas/water needs to be transported. The first and third level are coupled by $\tilde{x}\leq x$ and $\tilde{y}\leq y$, i.e., the third level may only use components that have been build in the first level, but it may also disable some. The second level represents worst-case failures (maximizing cost) of at most k components, where $z_i=1$ means that component $i\in[n]\coloneqq\{1,\dots,n\}$ fails. As already mentioned, x and $\tilde{x}$ represent the design decisions of components that are allowed to fail, enforced by $\tilde{x}\leq\mathbb{1}-z$. The components specified by y do not fail.

Model (2) involves some – more or less restrictive – assumptions:

1. The cost $F(x,y,u)$ is only based on the first level decisions, which means that costs in failing situations are ignored. This assumption is needed for our algorithmic approach and is reasonable for many applications. Note that this implies that the inner two levels are feasibility problems and (2) can be written as

$$\min_{(x,y,u)\in\mathcal{X}} \Big(F(x,y,u) + \max_{z\in\mathcal{Z}_k} \min_{(\tilde{x},\tilde{y},\tilde{u})\in\tilde{\mathcal{X}}} \big\{0 \,:\, \tilde{x}\leq x,\ \tilde{x}\leq\mathbb{1}-z,\ \tilde{y}\leq y\big\}\Big). \tag{3}$$

   Note that a design that is infeasible w.r.t. the second and third level automatically incurs infinite cost.

2. The second level failures are restricted to the set $\mathcal{Z}_k$, which is essential for the inequalities that we use in our algorithm. This means that all failures are equally likely and occur independently. Note that the distinction between x and y allows for some control and our setting covers many practical situations, or at least provides a conservative bound.

3. The third level designs may disable components that have been chosen by the first level. This makes the behavior of the solutions monotone, i.e., building more components will not change a feasible to an infeasible system. The validity of this assumption will depend on the particular system, see Remark 2 in Section 4.2 and the discussion in Section 4.3.

Our goal is the optimization of *resilient* technical systems, i.e., fault tolerant systems in the above sense. This wording comes fron mechanical engineering applications; see, e.g., [5], where resilience was extended to control uncertainty in technical systems.

1.1 Contribution and Outline

The main contributions of this paper are as follows:

- We develop a new algorithm for solving (2), based on a doubly nested Benders type method.
- The algorithm is generic in the sense that it is independent from a particular structure of the constraints and can, in particular, be applied to mixed-integer nonlinear problems.
- We demonstrate the algorithm on three nontrivial system design applications: water networks, truss topology optimization, and gas transport.

Our generic algorithm is based on two families of inequalities that allow for the reformulation of (2) as a single-level problem without additional variables. We then use a branch-and-cut approach that on the master level solves the corresponding (possibly nonlinear) system design problem and separates inequalities for integer design points. The separation problem is again tackled by a branch-and-cut method based on inequalities that are dynamically separated. To the best of our knowledge, our approach is different from the many different approaches to tackle problems similar to the above; see the literature discussion below.

It is important to note that we concentrate on the case in which the feasibility sets $\mathcal{X}$ and $\tilde{\mathcal{X}}$ are generic, i.e., we do not exploit their particular structure. In fact, since we deal with mixed-integer nonlinear and nonconvex problems, a practically feasible reformulation for the lower level based on optimality conditions seems to be out of reach. Moreover, we show that the decision version of (2) is on the third level of the polynomial hierarchy and thus extremely challenging. To provide solutions for practical instances, we essentially build on the (implicit) assumption that the number of failures k is small. Nevertheless, we show that our algorithm outperforms an algorithm based on the enumeration of failures.

We demonstrate the usefulness of our approach on three applications: water network, truss topology, and gas network design. The implementation of the algorithm is nontrivial and builds on the MINLP solver SCIP [26]. It can be easily combined with specific methods for the particular systems. For instance, for truss topology optimization, we employ SCIP-SDP [25, 57].

The three applications each highlight different use cases. For water networks, we concentrate on tree networks in high rise buildings and pumps may fail. The model of truss topology optimization is based on a mixed-integer semidefinite program (MISDP) and includes buckling constraints. Finally, the gas networks may include cycles and the focus is on the failure of pipes and the treatment of symmetry.

The paper continues as follows: We first give an overview of related work in Section 1.2 and introduce, for later comparison, a straight-forward enumeration solving scheme in Section 1.3. In Section 2, we investigate our problem in more detail. This includes a discussion of its computational complexity in Section 2.1, which suggests that it is inevitable to use a nested separation scheme. Furthermore, we provide and compare two reformulations of (2) in Section 2.2. In Section 2.3 we strengthen the formulation by consideration of symmetries within $\mathcal{X}$ and $\tilde{\mathcal{X}}$. Our algorithm is described in Section 3. The implementation of the branch-and-cut procedures are discussed in Section 3.1. We showcase the effectiveness of our algorithm on the three above mentioned examples: water networks (Section 4.1), truss topologies (Section 4.2), and gas networks (Section 4.3). The paper closes with conclusions in Section 5.

1.2 Literature Overview

The three levels in our problem can be interpreted as two players within a sequential game, commonly known as a Stackelberg game [61]. There exists extensive research which identifies these players as a defender and an attacker/interdictor within so-called defender-attacker-defender problems. Problems involving only two levels, e.g., attacker-defender problems, are referred as interdiction problems. In our context, the goal is to test whether a structure is resilient or to obtain the worst-case objective function. Applications for two and three level problems include electric grid planning [4, 11], which often investigates the N-K property – out of N components K may fail. Further applications arise from a military background [3], supply chain and facility location problems [53, 72], water networks [6], or truss topologies [24]. Also combinatorial optimization problems are investigated, e.g., the fortification of the traveling salesman problem by Lozano et al. [44], shortest path interdiction problem treated by Israeli and Woods [35], interdiction of a maximal flow by Wollmer [66], clique interdiction by Furini et al. [22] or Knapsack interdiction by Fischetti et al. [21].

Another separate area is the design of survivable networks, see, e.g., the overviews by Grötchel et al. [29] and Kerivin and Mahjoub [36]. A further viewpoint is the concept of bulk-robustness for combinatorial optimization problems, see Adjiashvili et al. [2, 1]. Here a set of failure scenarios is given and a solution is sought such that the properties of the underlying combinatorial problem, e.g., connectivity, is retained even after removing nodes belonging to failure scenarios. For more examples we refer to the survey on network resilience by Sharkey et al. [58] and the two surveys on interdiction by Smith and Lim [59] and Smith and Song [60].

Our problem can also be seen within the framework of adjustable robust optimization with integer variables, see e.g., Bertsimas et al. [9]. Interdiction problems belong to the more general setting of bilevel problems. Bilevel problems involving mixed integer linear constraints are, for example, solved by Tahernejad et al. [62] based on DeNegre [16], Fischetti et al. [20], and Tanınmış [63] using branch-and-cut approaches involving particular valid inequalities. Moreover, Mitsos [47], Kleniati and Adjiman [38] and Lozano and Smith [43] furthermore include nonlinear constraints. We refer to Kleinert et al. [37] for an informative overview.

There are two prominent solution strategies for defender-attacker-defender problems. The first uses convexity properties of the lowest level, allowing a one level formulation of the interdiction problem using KKT-conditions or the dual formulation of the lowest level. The second employs decomposition approaches and Benders type methods, which solve auxiliary problems to decide whether solution candidates are feasible or to derive valid cutting planes. Often both are combined. Since the former strategy is not applicable for our problem, we investigate the latter in more detail.

Zeng and Zhao [71] give one of the first generalized decomposition approaches. It solves two-stage robust optimization problems within a "column-and-constraint generation" framework. Here, one implicitly enumerates all second-level solutions and adds corresponding variables (columns) and constraints into the first level. To keep the problem size small, not all second-level solutions are considered, but an auxiliary problem is solved to find possibly missing ones. Note that the auxiliary problem contains two levels leading to an interdiction problem. This closely resembles the approach we present in Section 1.3 and use as a baseline for comparing our algorithm. This approach has been used several times and adapted to specific problems. Yuan, Zhao and Zeng [70] and also Wang et al. [65] use it to solve electric grid planning and unit commitment problems under the N-K property by reformulating the auxiliary problem using duality theory. Also [17] and [68] use solution algorithms based on the column-and-constraint generation framework to solve different defender-attacker-defender problems involving electric systems. Fang and Zio [19] use it to optimize resilient infrastructure systems involving a flow-based model.

Another decomposition approach is based on cutting planes often within a Benders' decomposition, similar to our solving scheme. Israeli and Wood [34, 35] identify a family of set covering inequalities to reformulated general interdiction problems into single level problems. The motivation of this "super-valid inequalities" is as follows, see also [67, 13]: In order for vector z to be a feasible (binary) interdiction vector, z has to cover each solution $\tilde{x}$ of the lowest level, i.e., $\tilde{x}^\top z \geq 1$ is valid. If it is possible to enumerate all $\tilde{x}$, one can directly solve the interdiction problem. Otherwise, as in the decomposition scheme above, only a small number of inequalities is used at the beginning and iteratively enlarged by solving an auxiliary problem. Church and Scaparra [14] use a similar approach to solve a facility location problem with fortification. The extension to three levels is done by nesting the decomposition, cf. [34], the applications on power network defense by Yao et al. [69] and network resilience by Ghorbani-Renani et al. [27]. Lozano and Smith [42] combine the covering constraints with sampling solutions of the lowest level.

This decomposition approach is close to our approach. We also find covering formulations for our problem, which we use in a nested decomposition. The main difference is the special structure of our interdiction problem, which specifies at most k failures to occur uniformly distributed. In contrast to the mentioned literature, this allows us to formulate the covering inequalities of the three level problem explicitly in terms of the recourse set $\tilde{\mathcal{X}}$.

1.3 Standard Algorithmic Approaches

In this section, we discuss two standard algorithmic approaches to solve Problem (3). These will later be used as a comparison baseline.

Since there are only finitely many failure scenarios z, a common approach to solve such three-level problems, is a reformulation as a single level MINLP. This approach is akin to a deterministic equivalent in stochastic programming. The model includes variables $\tilde{x}^z$, $\tilde{y}^z$, and $\tilde{u}^z$ for each failure scenario $z \in \mathcal{Z}_k$ and coupling constraints to link the additional variables with x, y, and z. Then the *scenario formulation (SF)* is given by

$$\begin{aligned} \min\ & F(x,y,u) \\ \text{s.t.}\ & \tilde{x}^z \leq x, && \forall z \in \mathcal{Z}_k, \\ & \tilde{y}^z \leq y, && \forall z \in \mathcal{Z}_k, \\ & \tilde{x}^z \leq \mathbb{1} - z, && \forall z \in \mathcal{Z}_k, \\ & (\tilde{x}^z, \tilde{y}^z, \tilde{u}^z) \in \tilde{\mathcal{X}}, && \forall z \in \mathcal{Z}_k, \\ & (x,y,u) \in \mathcal{X}. \end{aligned} \tag{4}$$

The problem can in principle be solved by standard MINLP solvers; we will call this the *static scenario approach*. However, due to the potential blowup in additional variables and constraints, SF can only be solved in practice for small instances and small k.

An alternative is the *dynamic scenario approach* (dynamic SA), in which the failure scenarios are only partially enumerated to obtain relaxations. By dynamically adding further failure scenarios and resolving the relaxations, the dual bounds become gradually tighter. We will describe this dynamic scenario formulation in the following. Consider the problem to find a failure scenario, which should be added to tighten the relaxations. Given a solution (x^*, y^*) to the previous relaxation this *adversarial problem (AP)* can be stated as a two level problem

$$\max_{z \in \mathcal{Z}_k} \min_{(\tilde{x},\tilde{y},\tilde{u}) \in \tilde{\mathcal{X}}} \left\{ 0 : \tilde{x} \leq x^*,\ \tilde{y} \leq y^*,\ \tilde{x} \leq \mathbb{1} - z \right\}. \tag{5}$$

One way to solve this bilevel problem is to utilize the finite lower-level decision space and reformulate it as follows:

$$\begin{aligned} \min\ & \mathbb{1}^\top z \\ \text{s.t.}\ & \tilde{x}^\top z \geq 1, && \forall (\tilde{x}, \tilde{y}) \in \tilde{\mathcal{X}} \text{ with } \tilde{x} \leq x^*,\ \tilde{y} \leq y^*, \\ & z \in \{0,1\}^n. \end{aligned} \tag{6}$$

It is unlikely that the points in $\tilde{\mathcal{X}}$ can be computed in advance. Thus, this set-covering problem can not be solved directly. However, it can be solved within a branch-and-cut approach;

our algorithm uses a similar procedure below. The corresponding separation problem can be formulated as

$$\min_{(\tilde{x},\tilde{y},\tilde{u})\in\tilde{\mathcal{X}}} \left\{(z^*)^\top \tilde{x} \,:\, \tilde{x} \le x^*,\ \tilde{y} \le y^*\right\} \tag{7}$$

for a relaxation solution z^*. This separation problem is assumed to be solvable by general MINLP-techniques.

2 Resilience Optimization

To investigate the structure of Problem (3) in more detail and to encapsulate the difficulties due to the consideration of failures, we use the following definition to describe feasible points.

Definition 2.1. Design decisions $(x, y) \in \{0,1\}^{n+m}$ are *k-resilient* with respect to $\tilde{\mathcal{X}}$ if and only if for each $z \in \mathcal{Z}_k$ there exists $(\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}$ with $\tilde{x} \le x$, $\tilde{y} \le y$, and $\tilde{x} \le \mathbb{1} - z$.

It is immediate that this definition can be used to reformulate Problem (3) as

$$\min_{(x,y,u)\in\mathcal{X}} \left\{F(x,y,u) \,:\, (x,y) \text{ is } k\text{-resilient w.r.t } \tilde{\mathcal{X}}\right\}. \tag{8}$$

In this section we first consider the complexity of deciding whether design decisions are k-resilient. We then present two linear integer formulations to characterize them.

2.1 Complexity

The multi-level form of Problem (3) indicates that it is necessary to consider higher levels of the polynomial time hierarchy than the complexity class NP to characterize its computational complexity. For an introduction of the hierarchy we refer to Arora and Barak [7] or Papadimitriou [51]. The complexity class Σ_3^P is found on the third level of the hierarchy. It can be defined to be the collection of decision problems which can be written as a logical formula $\exists u^1\ \forall u^2\ \exists u^3\ P(u^1, u^2, u^3)$ for a Boolean predicate P, which can be evaluated in polynomial time of the size of the Boolean variables u^i.

We consider the membership problem of Problem (3) with only linear constraints and $m = 0$. We show that this problem is Σ_3^P-complete. This implies that the problem can not be formulated as a single or even two-level problem with polynomial size unless the polynomial time hierarchy collapses.

Definition 2.2. The *Linear Resilient Design Decision Problem (RES)* has as input $k \in \mathbb{N}$ together with linear systems $Ax \le \tilde{b}$ and $\tilde{A}\tilde{x} \le \tilde{b}$. The problem asks whether there exists $x \in \mathcal{X} = \{x \in \{0,1\}^n \,:\, Ax \le b\}$ which is k-resilient with respect to $\tilde{\mathcal{X}} = \{\tilde{x} \in \{0,1\}^n \,:\, \tilde{A}\tilde{x} \le \tilde{b}\}$.

To show that (RES) is Σ_3^P-hard, we will reduce a Σ_3^P-complete version of SAT to (RES), namely the quantified satisfiability problem with three alternations of quantifiers. An instance of Σ_3^P-SAT is given by a Boolean formula φ and asks whether

$$\exists u^1\ \forall u^2\ \exists u^3\ \varphi(u^1, u^2, u^3) = 1,$$

where u^1, u^2, u^3 represent Boolean (binary) variables. If we use the prenex normal form, (RES) has a similar structure:

$$\exists x\ \forall z\ \exists \tilde{x}\ (Ax \le b, \tilde{A}\tilde{x} \le \tilde{b}, \tilde{x} \le x, \tilde{x} \le \mathbb{1} - z) \vee \sum_{i=1}^{n} z_i \ge k + 1.$$

Note that the term $\sum_{i=1}^{n} z_i \ge k + 1$ makes sure that only up to k failures in z are taken into account.

The reduction of Σ_3^P-SAT to (RES) would be easy, if there would exist y-variables that cannot fail and thus could model first and third level decisions u^1 and u^3. Here, however, we are interested in the complexity of (RES), in which all binary decisions can fail.

Lemma 2.3. (RES) *is* Σ_3^P*-hard.*

Proof. Let an instance of Σ_3^P-SAT be given. We may assume that the dimensions of each of the three arguments of φ is N and that φ is given in conjunctive normal form, i.e., $\varphi(u^1, u^2, u^3)$ is equal to

$$\bigwedge_{j=1}^{M}\Big(\bigvee_{i\in C_j^1} u_i^1 \vee \bigvee_{i\in \overline{C}_j^1} \neg u_i^1 \vee \bigvee_{i\in C_j^2} u_i^2 \vee \bigvee_{i\in \overline{C}_j^2} \neg u_i^2 \vee \bigvee_{i\in C_j^3} u_i^3 \vee \bigvee_{i\in \overline{C}_j^3} \neg u_i^3\Big),$$

where M gives the number of conjunctions and C_j^i and $\overline{C}_j^i$ are index sets to indicate the occurrence of u^i and its negation in conjunction j, respectively.

Given φ, our corresponding instance of (RES) has $k = N$ and $7N$ variables at each level. The first level variables are denoted x^1, x^{-1}, x^2, x^{-2}, x^3, x^{-3}, and $x \in \{0,1\}^N$. The system $Ax \leq b$ is:

$$x^1 + x^{-1} = \mathbb{1},\ x^2 = \mathbb{1},\ x^{-2} = \mathbb{1},\ x^3 = \mathbb{1}, x^{-3} = \mathbb{1},\ x = \mathbb{1}.$$

Thus, x^1 and x^{-1} represent the first level Boolean decisions u^1 and $\neg u^1$. The remaining variables do not interfere with the other levels.

The second level variables are z^1, z^{-1}, z^2, z^{-2}, z^3, z^{-3}, and $z \in \{0,1\}^N$. We will show that we can concentrate on the case $z^2 + z^{-2} = \mathbb{1}$ and the remaining variables being 0.

The third level variables are $\tilde{x}^1$, $\tilde{x}^{-1}$, $\tilde{x}^2$, $\tilde{x}^{-2}$, $\tilde{x}^3$, $\tilde{x}^{-3}$, and $\tilde{x} \in \{0,1\}^N$. The corresponding system $\tilde{A}\tilde{x} \leq \tilde{b}$ consists of

$$\tilde{x} \leq \tilde{x}^2,\ \tilde{x} \leq \tilde{x}^{-2},\ \tilde{x}^3 + \tilde{x}^{-3} \leq \mathbb{1}, \tag{9}$$

$$1 - \sum_{i=1}^{N} \tilde{x}_i \leq \sum_{\ell=1}^{3}\Big(\sum_{i\in C_j^\ell} \tilde{x}_i^\ell + \sum_{i\in \overline{C}_j^\ell} \tilde{x}_i^{-\ell}\Big), \qquad \forall j \in [M]. \tag{10}$$

The idea is that $\tilde{x}_i^j = 1$ iff u_i^j is true and $\tilde{x}_i^{-j} = 1$ iff u_i^j is false ($\neg u_i^j$ is true). Note that due to the coupling constraints $\tilde{x} \leq x$, the first level decisions x^1 and x^{-1} imply that $\tilde{x}^1 + \tilde{x}^{-1} \leq 1$, but allow for $\tilde{x}^1 = \tilde{x}^{-1} = 0$. This, however, has no advantage for satisfying $\tilde{A}\tilde{x} \leq \tilde{b}$. We can therefore assume that $\tilde{x}^1 + \tilde{x}^{-1} = 1$. Moreover, if $\tilde{x}_i^2 = \tilde{x}_i^{-2} = 1$, setting variable $\tilde{x}_i = 1$ is possible. In this case, the Boolean formula constraint represented in (10) is always satisfied.

Further assumptions on the second level variables can be made as follows. Assume that the second level variables are chosen such that $z_i^2 + z_i^{-2} + z_i \neq 1$ for some $i \in [N]$. Since $k = N$, this implies there exists $r \in [N]$ (possibly $r = i$) with $z_r^2 + z_r^{-2} + z_r = 0$. One can thus set $\tilde{x}_r^2 = \tilde{x}_r^{-2} = \tilde{x}_r = 1$ in the third level. This relaxes the constraints in (10), and the system $\tilde{A}\tilde{x} \leq \tilde{b}$ becomes easily satisfiable by setting the remaining third level variables to 0. Thus, we can assume that $z_i^2 + z_i^{-2} + z_i = 1$. Furthermore, $z_i = 1$ implies $\tilde{x}_i = 0$. However, this case is already covered by $z_i^2 + z_i^{-2} = 1$, which by $x_i^2 \leq 1 - z_i^2$, $x^{-2} \leq 1 - z_i^{-2}$, and (9) implies $\tilde{x}_i = 0$. We can therefore assume that $z_i = 0$. Thus, only those second level configurations are relevant in which $z^2 + z^{-2} = \mathbb{1}$. Since $k = N$, this implies that we can assume $z^1 = z^{-1} = z^3 = z^{-3} = \mathbb{0}$. These arguments show that these are the only relevant second level decisions for (RES), since all other interdictions are feasible or are covered by these configurations.

Therefore (RES) reduces to the following problem:

$$\exists (x^1, x^{-1}) \in \{0,1\}^{2n}\ \forall (z^2, z^{-2}) \in \{0,1\}^{2n}\ \exists (\tilde{x}^3, \tilde{x}^{-3}) \in \{0,1\}^{2n}$$
$$x^1 + x^{-1} = \mathbb{1},\ z^2 + z^{-2} = \mathbb{1},\ \tilde{x}^3 + \tilde{x}^{-3} \leq \mathbb{1},$$

$$1 \leq \sum_{i\in C_j^1} x_i^1 + \sum_{i\in \overline{C}_j^1} x_i^{-1} + \sum_{i\in C_j^2} z_i^2 + \sum_{i\in \overline{C}_j^2} z_i^{-2} + \sum_{i\in C_j^3} \tilde{x}_i^3 + \sum_{i\in \overline{C}_j^3} \tilde{x}_i^{-3}, \qquad \forall j \in [M].$$

This is equivalent to the Σ_3^P-SAT instance. □

Note that k is part of the input of (RES). For fixed k, the scenario formulation (4), enumerates $\binom{n}{k}$ failure scenarios and would be a linear integer feasibility problem with polynomial size in the input, which is NP-complete. Furthermore, the coupling between the first and third level but also the constraints on the first level variables are relevant for the complexity status. Otherwise, the problem is up-monotone and has a solution if and only if there exists a solution with $x = \mathbb{1}$, turning the problem into a two-level problem. Note that the above result shows that the difficulties of our problem are not only based on the inclusion of nonlinear functions.

2.2 Reformulations

In the following we will present two different characterizations of k-resilient points, which can be used to formulate Problem (8) as an MINLP in the original space.

Lemma 2.4. *A point $(x, y) \in \{0,1\}^{n+m}$ is k-resilient for $\tilde{\mathcal{X}}$ if and only if it satisfies*

$$\alpha^\top x + (k+1)\beta^\top y \geq k+1 \tag{11}$$

for each $(\alpha, \beta) \in \mathbb{F} := \{(\alpha,\beta) \in \{0,1\}^{n+m} : \tilde{x}^\top\alpha + \tilde{y}^\top\beta \geq 1 \ \forall(\tilde{x},\tilde{y},\tilde{u}) \in \tilde{\mathcal{X}}\}$.

Proof. Assume (x, y) is k-resilient, but there exists $(\alpha, \beta) \in \mathbb{F}$ such that (11) is not satisfied, i.e., $\alpha^\top x + \beta^\top y \leq k$. Note that this implies $\beta^\top y = 0$. Set $z_i := \alpha_i x_i$ for $i \in [n]$. Clearly $z \in \mathcal{Z}_k$. Thus, there exists a vector $(\tilde{x},\tilde{y},\tilde{u}) \in \tilde{\mathcal{X}}$ with $\tilde{x} \leq x$, $\tilde{y} \leq y$, and $\tilde{x} \leq \mathbb{1} - z$. Since $z_i = 0$ if $x_i = 0$, the latter implies $\tilde{x} \leq x - z$. But then we obtain the contradiction

$$1 \leq \tilde{x}^\top\alpha + \tilde{y}^\top\beta \leq \sum_{i=1}^n (x_i - z_i)^\top\alpha_i + 0 = \sum_{i=1}^n (x_i - \alpha_i x_i)^\top \alpha_i = 0,$$

where the last equality follows from α being binary and thus $\alpha_i^2 = \alpha_i$.

Conversely, assume (x, y) is not k-resilient. Then there exists some $z \in \mathcal{Z}_k$ such that there exist no possible response in $\tilde{\mathcal{X}}$. Define $\alpha \in \{0,1\}^n$ by $\alpha_i := z_i$ if $x_i = 1$, otherwise set it to 1. Define $\beta := \mathbb{1} - y$. Then

$$\alpha^\top x + (k+1)\beta^\top y \leq \mathbb{1}^\top z + (k+1)(\mathbb{1} - y)^\top y \leq k + 0.$$

Thus, (11) is violated and it remains to check $(\alpha,\beta) \in \mathbb{F}$. Let $(\tilde{x},\tilde{y},\tilde{u}) \in \tilde{\mathcal{X}}$. We show by case distinction that $\tilde{x}^\top\alpha + \tilde{y}^\top\beta \geq 1$. In the first case there exists i such that $\tilde{x}_i \not\leq x_i$ or $\tilde{y}_i \not\leq y_i$. Thus, $\tilde{x}_i = 1$ and $x_i = 0$ and thus $\alpha_i = 1$ or $\tilde{y}_i = 1$ and $y_i = 0$ and thus $\beta_i = 1$ by the definitions of α and β. We conclude that $\tilde{x}_i\,\alpha_i = 1$ or $\tilde{y}_i\,\beta_i = 1$. In the other case, i.e., $\tilde{x} \leq x$ and $\tilde{y} \leq y$, there must exist $i \in [n]$ such that $\tilde{x}_i = z_i = 1$: otherwise, $\tilde{x} \leq \mathbb{1} - z$ and $(\tilde{x},\tilde{y},\tilde{u})$ would be a response to the assumed to be fatal failure z. We obtain $x_i = 1$ and thus $\alpha_i = 1$. Altogether $\tilde{x}_i\,\alpha_i = 1$. □

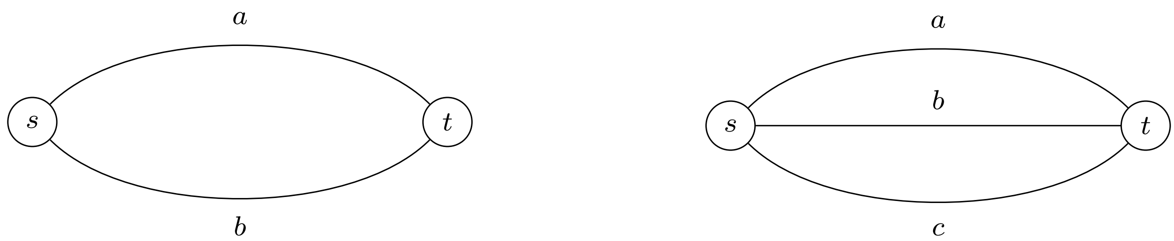
We derive another family of inequalities.

Lemma 2.5. *A point $(x, y) \in \{0,1\}^{n+m}$ is k-resilient for $\tilde{\mathcal{X}}$ if and only if it satisfies*

$$\alpha^\top x + \beta^\top y \geq 1 \tag{12}$$

for each $(\alpha,\beta,\gamma) \in \mathbb{F}_k$, where

$$\mathbb{F}_k := \{(\alpha,\beta,\gamma) \in \{0,1\}^{n+m} \times \mathcal{Z}_k : \tilde{x}^\top(\alpha+\gamma) + \tilde{y}^\top\beta \geq 1 \ \forall(\tilde{x},\tilde{y},\tilde{u}) \in \tilde{\mathcal{X}}\}.$$



(a) $E_1 = \{a\}$, $E_2 = \{b\}$. (b) $E_1 = \{a, b, c\}$.

Figure 1: Example graphs for the Fault-Tolerant Path Problem.

Proof. Let (x, y) be k-resilient and assume that $(\alpha, \beta, \gamma) \in \mathbb{F}_k$ is such that (12) is violated, i.e., $x^\top \alpha + y^\top \beta = 0$. Since $\gamma \in \mathcal{Z}_k$, γ is a possible failure scenario. Thus, there exists $(\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}$ with $\tilde{x} \leq x$, $\tilde{y} \leq y$ and $\tilde{x} \leq \mathbb{1} - \gamma$. The last inequality implies $\tilde{x}^\top \gamma = 0$. We also have $\tilde{x}^\top \alpha \leq x^\top \alpha = 0$ and $\tilde{y}^\top \beta \leq y^\top \beta = 0$. But then $(\alpha, \beta, \gamma) \notin \mathbb{F}_k$, because the corresponding inequality for $(\tilde{x}, \tilde{y}, \tilde{u})$ in $\mathbb{F}_k$ is violated – a contradiction.

Conversely, suppose (x, y) is not k-resilient. Thus there exists $z \in \mathcal{Z}_k$ such that for all $(\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}$ at least one of the inequalities $\tilde{x} \leq x$, $\tilde{y} \leq y$ and $\tilde{x} \leq \mathbb{1} - z$ is violated. Let $\alpha := \mathbb{1} - x$, $\beta := \mathbb{1} - y$, and $\gamma := z$. Then $\alpha^\top x + \beta^\top y = 0$ and it remains to show that $(\alpha, \beta, \gamma) \in \mathbb{F}_k$. This follows from

$$\tilde{x}^\top (\alpha + \gamma) + \tilde{y}^\top \beta = \tilde{x}^\top (\mathbb{1} - x + z) + \tilde{y}^\top (\mathbb{1} - y) \geq 1,$$

which holds by the inequality violation above. □

One immediate difference between the formulations is the clear advantage of $\mathbb{F}$ in Lemma 2.4 to produce less inequalities, whereas $\mathbb{F}_k$ in Lemma 2.5 leads to covering inequalities, which avoid the coefficient $k + 1$. We investigate further differences in the following example.

Example 2.6. We illustrate the inequalities on the Fault-Tolerant Path Problem considered by Adjiashvili et al. [1]. Given is an undirected graph $G = (V, E)$ with two distinct nodes s and t and a partition $E = E_1 \dot{\cup} E_2$. The goal is to find a set of edges $\hat{E} \subseteq E$, such that even after removing k edges in E_1 there still exists a path from s to t in $\hat{E}$. This can be formulated using binary variables x_e and y_e to signify whether an edge e in E_1 or E_2 is used, respectively. Then $\tilde{\mathcal{X}}$ consists of all points $(x, y) \in \{0, 1\}^{E_1 \times E_2}$ which describe an s-t path, i.e.,

$$\tilde{\mathcal{X}} = \operatorname{conv}(\{(x, y) \in \{0, 1\}^{E_1 \times E_2} : (x, y) \text{ form an } s\text{-}t \text{ path}\}).$$

Following Lemma 2.4, $(\alpha, \beta) \in \mathbb{F}$ if $\tilde{x}^\top \alpha + \tilde{y}^\top \beta \geq 1$ holds for all $(\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}$. This happens if and only if the edges $\{e \in E_1 : \alpha_e = 1\} \cup \{e \in E_2 : \beta_e = 1\}$ contain an s-t cut. By Condition (11), (x, y) is k-resilient, if it contains at least $k + 1$ edges in E_1 or one edge in E_2 from each s-t cut.

For $\mathbb{F}_k$ from Lemma 2.5, the inequality $\tilde{x}^\top (\alpha + \gamma) + \tilde{y}^\top \beta \geq 1$ means that there are k edges in E_1 which can be assigned to γ. Thus, according to Condition (12), (x, y) is k-resilient if and only if for each s-t cut, after excluding any k cut-edges in E_1, the edges given by (x, y) cover at least one of the remaining cut-edges.

Note that both types of inequalities follow from Menger's theorem, which is also used in survivable network design. In fact, neither dominates the other: Consider the two graphs in Figure 1 and $k = 1$. For the graph in Figure 1a, edge a is allowed to fail, whereas b can not. Thus, the 1-resilient solutions are given by edge b or both edges. In this case, $\mathbb{F}$ and $\mathbb{F}_k$ produce one undominated inequality each: $x_a + 2y_b \geq 2$ and $y_b \geq 1$, respectively.

In the second graph, all edges are allowed to fail. The nondominated inequalities in $\mathbb{F}$ correspond to the single s-t cut and is given by $x_a + x_b + x_c \geq 2$. For $\mathbb{F}_k$ we obtain more inequalities: $x_a + x_b \geq 1$, $x_a + x_c \geq 1$ and $x_b + x_c \geq 1$.

Strength of the Formulations. Since we use the inequalities in an LP or SDP-based branch-and-cut approach, it is interesting to compare the continuous relaxations of the formulations given by Lemma 2.4 and 2.5.

The example above shows that they are incomparable in general, e.g., the point $(x_a, y_b) = (1, 0.5)$ for the graph of Figure 1a violates Inequality (12) for $\mathbb{F}_k$ and can therefore be cut off. But the point satisfies the inequality from $\mathbb{F}$. Conversely, the point $(x_a, x_b, x_c) = (0.6, 0.6, 0.6)$ for the graph in Figure 1b violates the inequality from $\mathbb{F}$, but satisfies all inequalities from $\mathbb{F}_k$.

If all components are allowed to fail, the inequalities of Lemma 2.4 dominate the inequalities of Lemma 2.5.

Lemma 2.7. *For $m = 0$, we have*

$$\{x \in [0,1]^n : \alpha^\top x \geq k + 1 \;\forall \alpha \in \mathbb{F}\} \subseteq \{x \in [0,1]^n : \alpha^\top x \geq 1 \;\forall (\alpha, \gamma) \in \mathbb{F}_k\}.$$

This relation is strict in general.

Proof. Suppose that for $x \in [0,1]^n$ there exists $(\alpha, \gamma) \in \mathbb{F}_k$ with $\alpha^\top x < 1$. Define $\hat{\alpha} := \min(\alpha + \gamma, \mathbb{1})$. Then $\hat{\alpha} \in \mathbb{F}$, and

$$\hat{\alpha}^\top x \leq (\alpha + \gamma)^\top x < 1 + \gamma^\top x \leq 1 + \gamma^\top \mathbb{1} \leq 1 + k.$$

Thus, we would have also an violated inequality in $\mathbb{F}$. Strictness is shown by the example above. □

2.3 Symmetry Handling

One field of research to enhance branch-and-bound approaches is the identification and handling of symmetry in the problem, see, e.g., [45, 41, 31]. In the following, we will discuss a very simple case of symmetry in the design variables and how it can be handled in resilience problems.

Suppose some design decisions are fully symmetric, i.e., any permutation of a feasible solution leads to another feasible solution with the same objective value. This may happen, for example, with parallel identical connections (e.g., pipes, trusses, or links). To make this more precise, assume without loss of generality that the symmetric variables are $x_1, \dots, x_\ell$ for some $\ell \in [n]$. Then the full symmetric group S_ℓ is acting on the set $[\ell]$. Symmetry means that a point (x, y, u) belongs to $\mathcal{X}$ if and only if for each permutation $\sigma \in S_\ell$ there exists a point $(x', y, u') \in \mathcal{X}$ with $x'_i = x_{\sigma(i)}$ for $i \in [\ell]$. Furthermore, we demand the same for the lowest level, i.e., the permutation of feasible points in $\tilde{\mathcal{X}}$ can be made feasible. Note that we do not consider symmetry on y and that u' most likely coincides with u for some parts which are permuted according to σ, e.g., variables denoting volume flow of connections.

A simple possibility to handle such symmetry in an integer program is the addition of symmetry breaking constraints which allow only one representative, usually the lexicographically maximal:

$$x_i \geq x_{i+1}, \quad \forall i \in [\ell - 1]. \tag{13}$$

Suppose $(x, y, u) \in \mathcal{X}$ is k-resilient and let (x', y, u') be a feasible point after a permutation σ. Then the later point is also k-resilient: Since $\tilde{\mathcal{X}}$ is symmetric we can obtain a response for a failure scenario targeting (x', y) by permutation of a response to a permuted failure scenario for (x, y). Thus, the above symmetry breaking constraints can be added to $\mathcal{X}$ and we still retain k-resilient solutions.

Since we will need to optimize over the third level problem in the algorithm given below, we would preferably also add constraints like (13) to $\tilde{\mathcal{X}}$. However, if we do not change the definitions of the failure scenarios, this would cut off k-resilient solutions. Consider, e.g., $\tilde{\mathcal{X}} = \{(1, 0), (0, 1)\}$ and the 1-resilient point $x = (1, 1)$. Combining $\tilde{\mathcal{X}}$ with the symmetry breaking inequality $\tilde{x}_1 \geq \tilde{x}_2$ there would be no response for x to the failure scenario $z = (1, 0)$.

If we want to include symmetry breaking inequalities in $\tilde{\mathcal{X}}$, but retain an equivalent formulation to our original problem, the power of the failure scenarios has to be restricted. The following Lemma shows one way to incorporate symmetry breaking inequalities in the setting of Lemma 2.4. For the purpose of exposition, we concentrate on the case with only failable design decisions.

Lemma 2.8. *Let $m = 0$ and let $\tilde{\mathcal{X}}$ be symmetric for the first ℓ variables of x. Then $x \in \{0,1\}^n$ satisfying (13) is k-resilient with respect to $\tilde{\mathcal{X}}$ if and only if*

$$\alpha^\top x \geq k + 1$$

for each $\alpha \in \{0,1\}^n$ with $\alpha_i \leq \alpha_{i+1}$ for $i \in [\ell - 1]$ and $\tilde{x}^\top \alpha \geq 1$ for all $(\tilde{x}, \tilde{u}) \in \tilde{\mathcal{X}}$ with $\tilde{x}_i \geq \tilde{x}_{i+1}$ for $i \in [\ell - 1]$.

Proof. We first assume that x is k-resilient, but there exists α as in the above description, such that $\alpha^\top x \leq k$. We show that not only the lexicographically maximal elements of $\tilde{\mathcal{X}}$ are covered, but all, i.e., $\tilde{x}^\top \alpha \geq 1$ for all $(\tilde{x}, \tilde{u}) \in \tilde{\mathcal{X}}$. Then $\alpha \in \mathbb{F}$, which leads to a contradiction with Lemma 2.4.

Let $(\tilde{x}, \tilde{u}) \in \tilde{\mathcal{X}}$ and $\tilde{x}'$ be a permutation of $\tilde{x}$ such that $\tilde{x}'_i \geq \tilde{x}'_{i+1}$ for all $i \in [\ell - 1]$. Since $\tilde{\mathcal{X}}$ is symmetric, $(\tilde{x}', \tilde{u}') \in \tilde{\mathcal{X}}$ for some $\tilde{u}'$. Thus, $(\tilde{x}')^\top \alpha \geq 1$. If $\tilde{x}' = \tilde{x}$, then obviously $\tilde{x}^\top \alpha \geq 1$; therefore assume that $\tilde{x} \neq \tilde{x}'$ in the following. If $(\tilde{x}')^\top \alpha \geq 1$ because $\tilde{x}'_i \cdot \alpha_i = 1$ for some $i > \ell$, then also $\tilde{x}^\top \alpha \geq 1$ since $\tilde{x}_j$ and $\tilde{x}'_j$ coincide for $j > \ell$. Thus, assume that α covers $\tilde{x}'$ since there exists $i \in [\ell]$ with $\tilde{x}'_i \cdot \alpha_i = 1$. Then $x'_1 = \cdots = x'_i = 1$, because $\tilde{x}'$ is sorted non-increasingly. Because $\tilde{x}' \neq \tilde{x}$, there exists $j \in [\ell]$ with $j \geq i$ such that $\tilde{x}_j = 1$. Furthermore, α is sorted non-decreasingly. Thus, as $\alpha_i = 1$ and $i \leq j$, also $\alpha_j = 1$. Therefore α covers $\tilde{x}$.

Let x not be k-resilient. Thus, by Lemma 2.4 there exists an α such that $\alpha^\top x \leq k$ and $\tilde{x}^\top \alpha \geq 1$ for $\tilde{x} \in \tilde{\mathcal{X}}$. Permute α by $\sigma \in S_\ell$ to form α' with $\alpha'_i \leq \alpha'_{i+1}$ for $i \in [\ell - 1]$. Since x is sorted non-increasingly and α' non-decreasingly, we obtain from $\alpha^\top x \leq k$ also $(\alpha')^\top x \leq k$. Let $\tilde{x}' \in \tilde{\mathcal{X}}$. By symmetry of $\tilde{\mathcal{X}}$, we have that $\tilde{x}$ given by permuting $\tilde{x}'$ with σ^{-1} belongs to $\tilde{\mathcal{X}}$. But then

$$\sum_{i=1}^{n} \tilde{x}'_i \alpha'_i = \sum_{i=1}^{\ell} \tilde{x}'_i \alpha'_i + \sum_{i=\ell+1}^{n} \tilde{x}_i \alpha_i = \sum_{i=1}^{\ell} \tilde{x}_{\sigma(i)} \alpha_{\sigma(i)} + \sum_{i=\ell+1}^{n} \tilde{x}_i \alpha_i = \tilde{x}^\top \alpha \geq 1,$$

where we used that $\tilde{x}'_i$ and α'_i coincide with $\tilde{x}_i$ and α_i for $i > \ell$. □

Lemma 2.8 can be generalized to work for non-overlapping, independent symmetries in $\mathcal{X}$ and $\tilde{\mathcal{X}}$: the symmetry is given by an action of S_{i_r} on $I_r \subseteq [n]$ with $|I_r| = i_r$ for $r = 1, \ldots, R$, with pairwise disjoint sets I_r. Then the symmetry breaking inequalities for each I_r are added to the three levels. An example is given by sets of parallel pipes in a gas network optimization problem in Section 4.3.

In contrast, integrating symmetry into the scenario formulation (4) does not seem as easy. Replicating the idea to consider only sorted failure scenarios, i.e., $z \in \mathcal{Z}_k$ with $z_i \leq z_{i+1}$ for $i \in [\ell - 1]$, would make the second level too weak: For $k = 1$ we would only consider scenarios which have $z_i = 0$ for $i \in [\ell - 1]$. For the formulation from Lemma 2.5 we have similar problems. On the one hand, if we only consider cuts (α, γ) with $\gamma_i \leq \gamma_{i+1}$, we would again make the cuts too weak. On the other hand, if we follow the above Lemma and include the restriction $\alpha_i \leq \alpha_{i+1}$, we still can cut off k-resilient solutions, e.g., the above 1-resilient solution $x = (1, 1)$ with $\alpha = (0, 0)$ and $\gamma = (1, 0)$.

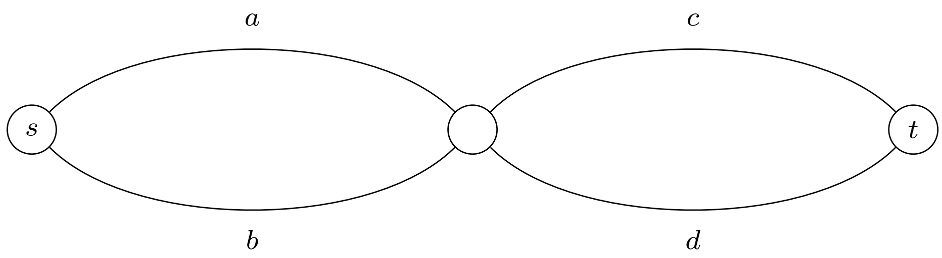


Figure 2: Example graph with $E = E_1 = \{a, b, c, d\}$.

3 Solution Algorithm and Implementation

In this section we propose a branch-and-cut approach so solve the reformulation of Problem (3) due to Lemma 2.4

$$\begin{aligned} \min\ & F(x, y, u) && && (14a)\\ \text{s.t.}\ & \alpha^\top x + (k+1)\beta^\top y \geq k+1, && \forall (\alpha, \beta) \in \mathbb{F}, && (14b)\\ & (x, y, u) \in \mathcal{X}. && && (14c) \end{aligned}$$

Of course, one could replace the *resilience inequalities* (14b) by the inequalities of Lemma 2.5. However, due to the domination result of Lemma 2.7 if $m = 0$ and the symmetry results, we decided to concentrate on this formulation. For a more concise exposition, we do not explicitly add symmetry breaking constraints; their addition is straight-forward.

We solve the formulation consisting of (14a) and (14c) using an MINLP solver. Let (x^*, y^*, u^*) be the solution at the current branch-and-cut node. If (x^*, y^*) is not integral or $(x^*, y^*, u^*) \notin \mathcal{X}$, we let the MINLP solver continue branching. Otherwise x^* and y^* are binary, and we solve a separation problem to check whether (14b) is violated and a cut should be added to the problem. This is sometimes called "lazy cut" approach.

Separation of (14b). We formulate the separation problem as

$$\begin{aligned} \min\ & (x^*)^\top \alpha + (k+1)\,(y^*)^\top \beta && && (15a)\\ \text{s.t.}\ & \tilde{x}^\top \alpha + \tilde{y}^\top \beta \geq 1, && \forall (\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}, && (15b)\\ & (\alpha, \beta) \in \{0,1\}^{n+m}. && && (15c) \end{aligned}$$

This problem is again solved with a branch-and-cut approach that separates the covering inequalities (15b). If the optimal value is smaller than $k + 1$, a violated cut is found. Since this separation problem is solved repeatedly (for different objective functions), we collect the separated covering inequalities and include them in further solves.

Remark 1. Given a violated failure scenario z, the proof of Lemma 2.4 yields the construction $\alpha := \min(\mathbb{1}, \mathbb{1} - x^* + z)$ and $\beta = \mathbb{1} - y^*$ of a violated resilience inequality. Thus, we could use the separation problem of the scenario formulation (6) instead. However, this does not necessarily give the best cuts within $\mathbb{F}$. Consider for example the Fault-Tolerant Path Problem for the graph of Figure 2 for $k = 1$. The only 1-resilient solution uses all four edges. A first solution in the branch-and-cut tree could be $x^* = (x_a^*, x_b^*, x_c^*, x_d^*) = (1, 0, 1, 0)$, i.e., the set containing edges a and c. Our separation algorithm for SF would find that $z = (1, 0, 0, 0)$ is a critical failure scenario. The corresponding inequality for this construction is then $x_a + x_b + x_d \geq 2$. However, as we have seen before, the coefficients for dominating inequalities in $\mathbb{F}$ correspond to s-t cuts in the graph. Thus, using z to construct the inequality does not yield the tighter inequality $x_a + x_b \geq 2$. Problem (15) allows in principle to obtain these tighter cut coefficients.

Separation of (15b). There are potentially exponentially many covering inequalities needed to describe Problem (15). Thus, we use another branch-and-cut scheme. Whenever a binary solution (α^*, β^*) during the solution process emerges, we solve

$$\min_{(\tilde{x},\tilde{y},\tilde{u})\in\tilde{\mathcal{X}}} (\alpha^*)^\top \tilde{x} + (\beta^*)^\top \tilde{y} \tag{16}$$

to check whether $(\alpha^*, \beta^*) \in \mathbb{F}$. Since a violated cut corresponds to a 0 objective value, we can presolve the problem and set $\tilde{x}_i$ to 0 when $\alpha_i = 1$.

3.1 Cut Improvement Strategies

The algorithm presented above can be improved as follows. When we are given a candidate (x^*, y^*) to check for resilience with Problem (15), we stop whenever a violated resilience cut is found, i.e., we do not necessarily find the tightest cut in the hope to save computational time. We furthermore fix $\alpha_i = 1$ if $x_i^* = 0$, which is possible since Problem (15) only consists of covering constraints. We also fix $\beta_i = 0$ if $y_i^* = 1$, which is possible since otherwise we would obtain a cut that is not violated in Problem (14). This does not change the objective value and reduces solving time. It, however, could cut off solutions (α, β), which correspond to tighter inequalities. Therefore, if (x^*, y^*) is not resilient, i.e., we computed (α^*, β^*) such that the corresponding inequality is violated, we devised two different strategies to hopefully obtain a tighter cut.

Cut Improvement by Resolving. The first strategy is to optimize the coefficients simultaneously. We change the objective coefficients in Problem (15) and allow only solutions which dominate the previously found cut:

$$\begin{aligned} \min \;& \mathbb{1}^\top \alpha + (k+1)\, \mathbb{1}^\top \beta \\ \text{s.t. } & \tilde{x}^\top \alpha + \tilde{y}^\top \beta \geq 1, \qquad \forall (\tilde{x}, \tilde{y}, \tilde{u}) \in \tilde{\mathcal{X}}, \\ & \alpha \leq \alpha^*,\ \beta \leq \beta^*, \\ & (\alpha, \beta) \in \{0,1\}^{n+m}. \end{aligned} \tag{17}$$

When solving this problem, the covering inequalities are separated using Problem (16). To control the additional work, which hopefully leads to better resilience cuts, we use limits on the number of separated cover cuts and the additional number of branch-and-cut nodes. If these limits are exceeded, we stop the improvement step and take the best cut we have found so far. In our computational experiments, see Section 4.1 and following, we compare different limits.

Greedy Cut Improvement. The second strategy passes through the indices for which α_i^* or β_j^* are 1 and tests via Problem (16) whether zeroing this coefficient leaves the cut feasible. If this is the case, we reduce its coefficient to 0, otherwise it remains 1. We determine the order of testing according to the covering score of the variables: the number of times the corresponding variable has value 1 in one of the solutions of $\tilde{\mathcal{X}}$ found when solving Problem (16). We test two variants below: Starting with the variables with the lowest ranking or with the highest ranked variables. The former has the aim to remain feasible, since less covering inequalities can be violated. The latter tries to set these possibly very impactful cut coefficients to zero.

4 Application Examples and Computational Experiments

In the following we present computational results on three classes of test instances for our implementation of the above solving scheme. We use a bugfix version of the MINLP solver

SCIP 7 [26]. Moreover, we use IPOPT 3.12 [64] as NLP-solver and CPLEX 12.10 as LP-solver. For the instances involving mixed integer semidefinite problems, we use a modified developer version of SCIP-SDP 3.3 [25, 57] which in turn uses MOSEK 9.2.35 [49] as SDP-solver. The source code and log files of the results can be obtained via the first author's webpage. All computations were performed single threaded on a Linux cluster with Intel Xeon E5 CPUs with 3.50 GHz, 10 MB cache, and 32 GB memory using a 1 hour time limit. We use the default settings of SCIP and SCIP-SDP for solving.

We implemented the algorithm in such a way that generic problems can be solved: The user specifies the set $\mathcal{X}$ including the objective function and the set $\tilde{\mathcal{X}}$ by two problem files, readable by SCIP. Furthermore, k and a list of variables names which specify x and y is given. This makes the adaption to different applications very easy.

In our tests we compare the following algorithms and settings:

- The static scenario approach (static SA) using the full scenario formulation (4).
- The dynamic scenario approach (dynamic SA) using (15) as separation problem and transforming violated cuts to violated scenarios using the transformation given in the proof of Lemma 2.4.
- Our algorithm with one choice of each of the two following possibilities:
  - Greedy strategy:
    - Not using the greedy strategy (–).
    - Greedy strategy with processing variables in increasing order of covered solutions (increasing ranking $\nearrow$).
    - Greedy strategy with processing variables in descreasing order of covered solutions (decreasing ranking $\searrow$).
  - Resolving strategy:
    - Not using the resolving strategy (–).
    - Using the resolving strategy with a node limit of 1 to solve Problem (15) and stopping the resolve after five separated cuts (5).
    - Using the resolving strategy with a node limit of 1 to solve Problem (15) without limiting the separation of cuts (∞).

4.1 Design of Resilient Water Network Designs

Our first application has the goal to find a selection of pipes and pumps such that a given volume flow and pressure demand can be satisfied with minimal investment and running cost, and such that a given volume flow and pressure demand can be ensured even though k of the pumps can fail.

Water network design using MINLP is, for example, investigated in Morsi et al. [48], Gleixner et al. [28] and D'Ambrosio et al. [18]. The integration of pump failures into MINLP models has been considered in [6].

To describe our model, we use a directed graph (V, A) and a finite set of load scenarios S. For each node v and each scenario $s \in S$ we have given lower and upper bounds on volume flow demand $\underline{q}_v^s$, $\overline{q}_v^s$ and pressure $\underline{p}_v^s$, $\overline{p}_v^s$. Furthermore, there are lower and upper bounds $\underline{q}_a$, $\overline{q}_a$ on the volume flow for each arc a. The arcs are divided into passive and active arcs $A = A^p \dot{\cup} A^a$. To connect the network, passive arcs can be used for which the pressure difference between the connected nodes is zero, i.e., we neglect friction. The active arcs can be used to increase the pressure along the arc. This pressure difference is bounded by two quadratic polynomials $\underline{P}_a(\cdot)$, $\overline{P}_a(\cdot)$ in the volume flow over the arc. Increasing the pressure is reflected in the objective by including energy costs according to a cubic polynomial $E_a(q, \Delta p)$

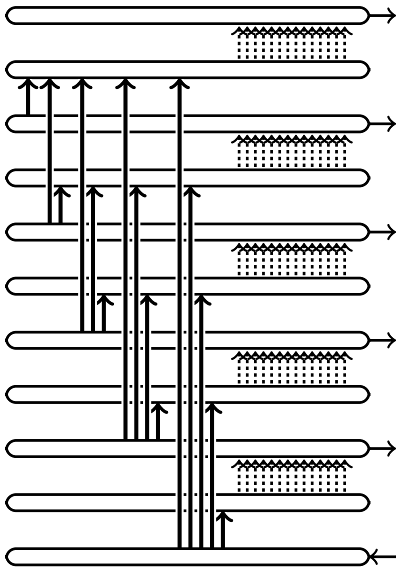


Figure 3: Graph of possible design choices for a water network depicting the high-rise testrig. The volume flow demand of the floors can be provided by connecting them to the basement by passive arcs (⟶). On each floor active arcs (⇢) can be used to increase the pressure.

in the volume flow q and pressure difference Δp, which is weighted by C_s for a given scenario s. The layout is then determined by binary variables x_a and y_a with cost C_a to determine the construction of active and passive arcs a, respectively. Furthermore, binary variables w_a^s for arcs in A^a determine the arc usage in scenario s.

The full MINLP to describe the system design optimization problem, without failure consideration, i.e., the model corresponding to Problem (1), is given below. Here $\delta^-(v)$ and $\delta^+(v)$ denote the incoming and outgoing arcs of node v, respectively.

$$
\begin{aligned}
\min\quad & \sum_{a\in A^a} C_a x_a + \sum_{a\in A^p} C_a y_a + \sum_{a\in A^a}\sum_{s\in S} C_s E_a(q_a^s, \Delta p_a^s)\, w_a^s \\
\text{s.t.}\quad & \underline{q}_v^s \le \sum_{a\in\delta^-(v)} q_a^s - \sum_{a\in\delta^+(v)} q_a^s \le \overline{q}_v^s, && \forall v\in V, s\in S,\\
& \underline{q}_a^s\, y_a \le q_a^s \le \overline{q}_a^s\, y_a, && \forall a\in A^p, s\in S,\\
& \underline{q}_a^s\, w_a^s \le q_a^s \le \overline{q}_a^s\, w_a^s, && \forall a\in A^a, s\in S,\\
& (p_v^s - p_u^s)\, y_a = 0, && \forall a=(u,v)\in A^p, s\in S,\\
& (p_v^s - p_u^s)\, w_a^s = \Delta p_a^s, && \forall a=(u,v)\in A^a, s\in S,\\
& w_a^s \le x_a, && \forall a\in A^a, s\in S,\\
& \underline{P}_a(q_a^s) \le \Delta p_a^s \le \overline{P}_a(q_a^s), && \forall a\in A^a, s\in S,\\
& q^s\in\mathbb{R}^A,\ p^s\in[\underline{p}^s,\overline{p}^s],\ \Delta p^s\in\mathbb{R}^{A^a}, && \forall s\in S,\\
& w^s\in\{0,1\}^{A^a}, && \forall s\in S,\\
& x\in\{0,1\}^{A^a},\ y\in\{0,1\}^{A^p}.
\end{aligned}
\tag{18}
$$

We generated our instance from the design of a modular testrig given by Müller et al. [50] to experimentally verify optimization models of a high-rise water network with five load scenarios. The underlying graph is illustrated in Figure 3. To implement tree conditions additional cardinality constraints $\sum_{a\in A^p\cap\delta^-(v)} x_a \le 1$ on the passive arcs are included in the above problem.

We vary k from 1 to 3. Furthermore, we consider the set $\tilde{\mathcal{X}}$ as given by the constraints of Problem (18) for only one load scenario and its corresponding volume flow and pressure

Table 1: Water network design test results for different settings and algorithms.

cut improvement strategy		time	# nodes	# solved
greedy	resolving			
–	–	2203.57	118740.3	9
–	5	1102.90	63039.0	17
–	∞	958.86	50276.2	18
$\searrow$	–	1272.37	71000.8	17
$\searrow$	5	988.91	56092.1	18
$\searrow$	∞	954.40	50012.6	18
$\nearrow$	–	958.00	55242.1	18
$\nearrow$	5	930.78	53710.4	18
$\nearrow$	∞	965.30	51232.9	18
static SA		3600.00	3050.8	0
dynamic SA		3136.72	876.1	3

bounds. This load scenario is considered in six different variants, in which the volume flow demand is scaled from 50 % up to 100 % of the maximal value in case of no failure.

Our implementation also separates particular perspective cuts, which can be derived as a special case of the approach of Bestuzheva et al. [10]. These cuts work with the on/off constraints on the active arcs and the nonlinear connection between volume flow and pressure difference. We heuristically determine valid linear underestimates on the cubic energy function $E(q, \Delta p)$ and combine them with the indicator variable w.

Table 1 shows a summary of our computational experiments for different settings and algorithms listed in the beginning of Section 4 over the set of 18 instances. For each setting we give the shifted geometric mean[1] of the total running time in seconds and the sum of all branch-and-cut nodes visited in all levels; the shift is 1 and 100, respectively. Furthermore, the number of solved instances in the testset is given. If we disregard failures, we can solve the problem within 180 s, while the consideration of failures increases the solving time significantly. One can see that the improvement strategies work: they allow to solve all instances within the time limit, whereas we only solve half of the instances without cut improvement. If we do not use a greedy strategy, it is beneficial to use more time on resolving. This also holds in combination with the greedy strategy with decreasing rank. For a combination with increasing greedy strategy this does not hold.

We also tested the scenario formulation (4) as a static and also dynamic approach, where we use (15b) to find a violated failure scenario. Both approaches are inferior to our algorithm. None of the instances can be solved within the time limit with the static version, the dynamic version can solve three. Whereas the formulation of the $k = 0$ problem and a formulation of $\tilde{\mathcal{X}}$ has around 3000 variables and 6000 constraints, the formulation with all scenarios explicitly added has around 30 500 variables and 60 000 constraints after presolving for $k = 1$.

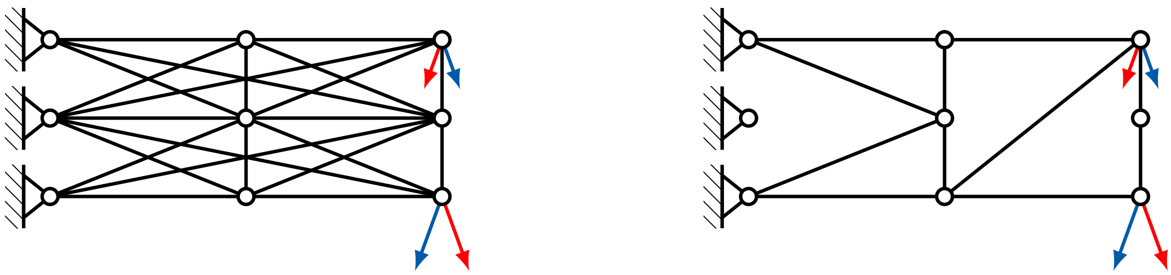
For the best setting, which tries to reduce coefficients with the smallest covering rank first and resolving with at most 5 separated covering cuts, Table 2 breaks down the solving statistic for each instance. We show for each instance the time for solving the whole problem, the time to separate our resilience cuts (time $\mathbb{F}$), i.e., the aggregated time we need to solve Problem (15), and the aggregated time to solve Problem (16) (time $\tilde{\mathcal{X}}$). The latter time includes the time to separate the covering inequalities when solving Problem (15), but also the time within the cut improvement strategies. The table furthermore includes the number of times we solve the two separation problems (# solves $\mathbb{F}$, # solves $\tilde{\mathcal{X}}$). Lastly, we present the aggregated number of computed cutting planes ($|\mathbb{F}|$ and $|\tilde{\mathcal{X}}|$).

We observe a clear trend in increasing solving time for increasing k. The time to find violated resilience cuts is relative low and at most 11 need to be added at the first level. It seems that SCIP has to invest a lot of time to resolve the nonlinearities on the upper level.

[1] The shifted geometric mean of $x_1, \ldots, x_n$ with shift s is given by $(\prod_{i=1}^{n}(x_i + s))^{\frac{1}{n}} - s$.

Table 2: Detailed water network test results using the increasing greedy and resolving strategy with 5 cuts. Column "$\tilde{\mathcal{X}}$ %" indicates the percentage volume flow demand after failures.

k	$\tilde{\mathcal{X}}$ %	time	time $\mathbb{F}$	time $\tilde{\mathcal{X}}$	# solves $\mathbb{F}$	# solves $\tilde{\mathcal{X}}$	$\lvert\mathbb{F}\rvert$	$\lvert\tilde{\mathcal{X}}\rvert$
1	50	569.32	10.53	9.60	217	309	4	112
1	60	439.48	9.39	8.52	205	299	3	116
1	70	613.53	15.52	12.69	837	284	3	102
1	80	566.67	38.00	36.01	632	360	5	127
1	90	580.77	30.29	29.51	267	286	4	126
1	100	977.42	102.35	91.39	3360	735	11	231
2	50	1070.33	7.86	6.39	455	258	3	111
2	60	638.84	11.79	10.31	410	312	4	126
2	70	912.18	30.03	23.07	2236	483	6	162
2	80	716.17	39.97	35.18	1518	441	7	162
2	90	1018.03	164.66	162.34	751	729	9	248
2	100	1148.64	117.43	111.63	1327	735	9	247
3	50	1741.23	17.26	10.72	2300	322	4	127
3	60	818.14	14.26	12.42	434	409	4	156
3	70	878.20	39.22	33.77	1353	694	9	225
3	80	2890.49	241.88	57.80	52910	560	8	201
3	90	2590.94	326.35	41.59	53787	401	6	155
3	100	1135.28	245.65	78.13	39087	557	8	214



(a) Ground structure.

(b) Example solution.

Figure 4: Example truss instance with six free nodes and two forces attacking the truss and a solution.

Generally, most of the time needed to solve the separation problem (15) is spent in separation of covering inequalities over $\tilde{\mathcal{X}}$. It turns out that the three instances for which this relation does not hold ($k = 3$ with volume flow at least 80 %) are infeasible. Here the number of solved separation problems is a magnitude larger than for the remaining instances. The number of solves over $\tilde{\mathcal{X}}$, however, remains the same. This is due to the fact that we check previously found solutions of $\tilde{\mathcal{X}}$ before we start solving the lower separation problem (16) over $\tilde{\mathcal{X}}$. We note that the solution over $\tilde{\mathcal{X}}$ is relatively fast: on average a fraction of a second is needed per solve.

4.2 Truss Topology Design with Bar Failures

As another test case we consider the design of light-weight truss topologies under consideration of buckling constraints. We follow Mars [46], Kočvara [40], and Gally [23], who formulate the problem to choose discrete beam sizes with respect to uncertain loads as a mixed integer semidefinite program (MISDP) and follow [24] for the introduction of buckling constraints into the MISDP.

Given is a so-called ground structure as an undirected graph (V, E). The goal of the problem is to choose bars, i.e., edges of E, such that the resulting truss is stable, see Figure 4. In more detail, we consider n_f free nodes $V_f \subseteq V$, the remaining nodes are fixed. In dimension two this leads to $d_f = 2n_f$ degrees of freedom. We assume that the truss is stable if its compliance is bounded by $C_{\max}$ for a finite number of scenarios given by forces $f_s \in S \subset \mathbb{R}^{d_f}$. This can be modeled using semidefinite constraints. Together with binary variables x_e for $e \in E$ one obtains a MISDP. Additionally, we consider an extension of the model containing buckling of bars, which happens due to tension or compression, i.e., some bar forces exceed critical values. Using actuators the critical buckling load can be increased. To keep costs under control, only a limited number r should be used. To determine the buckling forces, indicator constraint are used. The whole model is given by

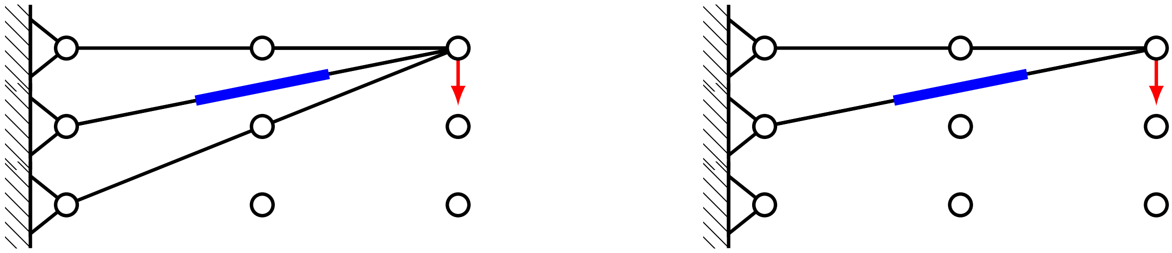
$$
\begin{aligned}
\min \ & \sum_{e \in E} \ell_e x_e \\
\text{s.t.} \ & \begin{pmatrix} 2C_{\max} & f_s^\top \\ f_s & A(x) \end{pmatrix} \succeq 0, && \forall f_s \in S \\
& B\, q_s = f_s, && \forall f_s \in S, \\
& -\frac{\pi E a^2}{4\ell_e^2} x_e - \rho\, y_e \leq (q_s)_e \leq R_{p;0,2}\, a\, x_e, && \forall f_s \in S, e \in E, \\
& \sum_{e \in E} y_e \leq r, \\
& y_e \leq x_e, && \forall\ e \in E, \\
& x_e = 1 \Rightarrow (q_s)_e = \frac{Ea}{\ell_e} b_e^\top u_s, && \forall f_s \in S, e \in E, \\
& x \in \{0,1\}^E,\ y \in \{0,1\}^E, \\
& q_s \in \mathbb{R}^E,\ u_s \in \mathbb{R}^{d_f}, && \forall f_s \in S,
\end{aligned}
\tag{19}
$$

where the topology is determined by the bars choosen by x and the usage of actuators given by y. The variables u represent node displacements which are needed to compute the barforces q. We denote the length of a bar e by l_e, the diameter of the equally thick bars by a, Young's modulus of the used material by E, and $A(x)$ denotes the stiffness matrix of the truss depending which bars x are used. Furthermore, b_e is the column corresponding to bar e in the geometry matrix of the truss. The proportional limit $R_{p;0,2}$ is used to express a bound for barforces under tension. Lastly, ρ is the increase of the critical buckling load due to an active actuator. For more details we refer to [24].

In our setting, we assume that failures of bars are possible, whereas actuators are persistent. Thus, we seek to find a light-weight truss x and actuators y such that the failure of k bars of x still leads to a stable truss. We demand that the remaining truss has to have the same maximal compliance $C_{\max}$ for the same given forces as in the nominal level, i.e., $\mathcal{X} = \tilde{\mathcal{X}}$, and the sets correspond to the feasible region of Problem (19) including all scenarios in S.

Example 4.1. Interestingly, the presented problem is not up-monotone with respect to x due to the buckling constraints. Adding bars can redistribute bar forces in such a way that the critical values are exceeded. An example is given in Figure 5. The truss shown in Figure 5a is not feasible, since the bar forces of the lowest two bars exceed their critical limit. Removing these two bars, see Figure 5b, the bar force in one of the remaining bars is increased, but does not exceed the critical bar force, since the bar is equipped with an actuator.

Remark 2. In [24], buckling was introduced into a resilience model. This model does not allow to disable bars after failure of other bars, in contrast to our definition, which allows to make the above example feasible. If the system was monotone, both approaches would lead to the same solutions. One could question the practicability of our framework: Does one need to manually remove these bars? Another interpretation, however, is that a k-resilient



(a) Infeasible truss. (b) Feasible truss.

Figure 5: Infeasible (5a) and feasible (5b) truss, showing that the problem is not monotone in general. The blue labeled bar is equipped with an actuator, which increases the critical buckling force.

Table 3: Truss topology design test results for different settings and algorithms.

cut improvement strategy		time	# nodes	# solved
greedy	resolving			
–	–	212.29	22481.7	19
–	5	22.49	3127.1	28
–	∞	24.21	3431.3	28
↘	–	22.83	3188.6	28
↘	5	22.38	3194.7	28
↘	∞	25.18	3580.5	28
↗	–	34.25	4897.9	28
↗	5	25.26	3646.2	28
↗	∞	25.42	3684.4	28
static SA		48.61	102.9	23
dynamic SA		300.20	4819.2	16

truss is still stable after the failure of a given number of bars and the cascading failure of further bars due to buckling. In this sense, we find sensible solutions.

Our instances have different sizes to form a cantilever as in Figure 4 with $2{\times}3$, $3{\times}2$, $2{\times}4$, $4{\times}2$ and $3{\times}3$ nodes. Furthermore, we consider three different bar-diameters. Two actuators may be placed. To generate solution candidates, we also use a heuristic, which works on a given branch-and-cut node, sets all design variables x to their current upper bound. To obtain solution values for the remaining free variables, we solve a a single SDP.

We reuse the settings investigated before. The results are given in Table 3. Again, the effect of the cut improvement is considerable. The best settings can solve 28 out of 30 instances in one hour, whereas without improving the cuts only 19 can be solved. No strategy dominates all others. In contrast to the water network instances, the static scenario approach (4) solves a considerable number of instances within the time limit. It solves 23 instances and performs better than our algorithm without cut improvement. The dynamic scenario approach performs worst. We highlight the small number of visited branch-and-cut nodes for the static method.

Table 4 gives statistics for each instance for the best setting according to average time, which improves the cuts via the decreasing greedy strategy, but does not use the resolving strategy. Compared to the water network testset, much more time is used for constraint separation. This is also reflected in the far larger number of separated inequalities and the larger number of solutions of $\tilde{\mathcal{X}}$ which are encountered.

Table 4: Detailed truss topology design test results using the decreasing greedy but no resolving strategy. Columns "Ground" and "diam" denote the ground structure and the diameter used of the instance.

k	ground	diam	time	time $\mathbb{F}$	time $\tilde{\mathcal{X}}$	# solves $\mathbb{F}$	# solves $\tilde{\mathcal{X}}$	$\lvert\mathbb{F}\rvert$	$\lvert\tilde{\mathcal{X}}\rvert$
1	2×3	50	0.96	0.48	0.39	55	62	9	16
1	2×3	150	1.59	1.00	0.65	209	216	10	52
1	2×3	250	1.71	1.09	0.67	215	245	11	62
1	2×4	50	52.00	38.79	36.97	593	724	9	228
1	2×4	150	12.63	8.07	5.86	652	1093	21	353
1	2×4	250	13.00	7.22	4.97	905	1190	25	355
1	3×2	50	0.52	0.34	0.26	25	49	5	13
1	3×2	150	1.72	0.96	0.60	168	185	9	51
1	3×2	250	1.92	0.98	0.62	186	185	9	44
1	3×3	50	3600.00	3480.86	3448.27	2984	5884	148	2237
1	3×3	150	151.00	46.85	30.85	2978	4129	31	1318
1	3×3	250	186.76	47.30	28.20	3266	4492	25	1432
1	4×2	50	182.82	175.22	173.04	577	859	54	309
1	4×2	150	19.26	9.01	7.06	573	862	14	277
1	4×2	250	22.41	8.73	6.89	603	903	16	289
2	2×3	50	0.70	0.37	0.33	24	45	4	12
2	2×3	150	2.92	2.12	1.37	309	364	10	90
2	2×3	250	2.45	1.79	1.18	233	312	7	81
2	2×4	50	76.78	66.09	64.59	455	646	13	201
2	2×4	150	21.24	13.07	8.00	1380	1582	22	437
2	2×4	250	24.32	14.09	8.14	1681	1791	24	506
2	3×2	50	0.83	0.55	0.42	55	71	6	19
2	3×2	150	2.43	1.67	1.12	195	260	14	67
2	3×2	250	3.87	2.57	1.71	265	367	11	110
2	3×3	50	3600.00	3526.87	3497.77	2111	4428	14	1943
2	3×3	150	282.95	127.26	60.02	5903	7727	23	2397
2	3×3	250	535.04	237.42	73.06	9918	11552	32	3523
2	4×2	50	39.93	36.74	36.11	189	359	10	117
2	4×2	150	168.19	147.17	133.44	1697	3501	52	1289
2	4×2	250	65.02	39.45	30.90	1305	2585	35	955

4.3 Gas Networks

The third application is the design of stationary gas networks. Here, we are given a directed graph (V, A), which provides the potential connections to be built. We will keep the following description very brief and refer to the literature for more details, e.g., [52, 39]. Topology optimization of gas networks is discussed, for instance, in [33, 32, 12, 56].

There is a pressure p_v for each node $v \in V$ in the network and a mass flow rate q_a for each arc $a = (u, v) \in A$; we have $q_a > 0$ if there is flow from u to v and $q_a < 0$ if there is flow in the opposite direction. Moreover, for each node $v \in V$, there is a fixed given outflow/inflow q_v. Note that here we only consider a single load scenario.

We distinguish several types of arcs $A = A^{\text{pi}} \dot\cup A^{\text{va}} \dot\cup A^{\text{cs}} \dot\cup A^{\text{sp}}$. For a *pipe* arc $a = (u, v) \in A^{\text{pi}}$, we use the so-called Weymouth equations $p_u^2 - p_v^2 = \beta_a \, q_a |q_a|$. The coefficients $\beta_a > 0$ incorporate information like length, diameter, and resistance; in this paper we consider them to be constant. These equations imply that there is flow from the node with higher pressure to the one with lower pressure. Moreover, for simplicity, we assume that there are no height differences. For a *valve* arc $a = (u, v) \in A^{\text{va}}$, there is a binary variable x_a, which is 0 iff the valve is closed. In this case, the flow q_a is 0 and the pressures at the end nodes are decoupled. If $x_a = 1$ the valve is open and $p_u = p_v$ has to hold. For a *compressor* arc $a \in A^{\text{cs}}$, there is a binary variable w_a that is 1 iff we are allowed to increase the pressure. Here, we use a very simple model that just allows a positive pressure difference within bounds $0 < \underline{\Delta}_a \leq p_v - p_u \leq \overline{\Delta}_a$. Note that compressors can only be used in forward direction. Valves in parallel to a compressor may allow a bypass. This avoids switching a compressor on or allows to have flow in the opposite direction. A *short pipe* arc $(u, v) \in A^{\text{sp}}$ requires that the pressures at the end nodes are equal. Additionally, we can have control valves and resistors, but we will not describe them in detail here.

Using lower and upper bounds $0 \leq \underline{p}_v \leq p_v \leq \overline{p}_v$ for the pressures and $\underline{q}_a \leq q_a \leq \overline{q}_a$ for the flows, respectively, valves and compressors are easy to model. A design problem can be obtained by adding a valve to each potential component of the network. The objective is then to minimize the number of such valves that are open. For simplicity, we assume that all of these valves/components are allowed to fail. This yields the following model:

$$
\begin{aligned}
\min \quad & \sum_{a \in A^{\text{va}}} C_a \, x_a \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} q_a - \sum_{a \in \delta^-(v)} q_a = q_v, && \forall v \in V, \\
& p_u^2 - p_v^2 = \beta_a \, q_a \, |q_a|, && \forall a = (u, v) \in A^{\text{pi}}, \\
& 0 \leq q_a \leq \overline{q}_a \, w_a, && \forall a = (u, v) \in A^{\text{cs}}, \\
& p_v - p_u + (\overline{p}_v - \underline{p}_u - \overline{\Delta}_a) \, w_a \leq \overline{p}_v - \underline{p}_u, && \forall a = (u, v) \in A^{\text{cs}}, \\
& p_v - p_u + (\underline{p}_v - \overline{p}_u - \underline{\Delta}_a) \, w_a \geq \underline{p}_v - \overline{p}_u, && \forall a = (u, v) \in A^{\text{cs}}, \\
& \underline{q}_a \, x_a \leq q_a \leq \overline{q}_a \, x_a, && \forall a = (u, v) \in A^{\text{va}}, \\
& p_v - p_u + (\overline{p}_v - \underline{p}_u) \, x_a \leq \overline{p}_v - \underline{p}_u, && \forall a = (u, v) \in A^{\text{va}}, \\
& p_v - p_u + (\underline{p}_v - \overline{p}_u) \, x_a \geq \underline{p}_v - \overline{p}_u, && \forall a = (u, v) \in A^{\text{va}}, \\
& p_u = p_v, && \forall a = (u, v) \in A^{\text{sp}}, \\
& \underline{q}_a \leq q_a \leq \overline{q}_a, && \forall a \in A, \\
& \underline{p}_v \leq p_v \leq \overline{p}_v, && \forall v \in V, \\
& w \in \{0,1\}^{A^{\text{cs}}}, \; x \in \{0,1\}^{A^{va}}.
\end{aligned}
\tag{20}
$$

In addition we use the methods from [30], which exploit the fact that gas flows are acyclic, i.e., there is no flow along a cycle. The corresponding model adds binary variables for each arc determining the flow direction and cycle inequalities that avoid such cycles. For more details, see [30].

We highlight two properties that distinguish the nonlinear behavior as in gas networks from a linear one.

1. In classical network design with linear flows and positive costs, every optimal solution is a tree, a so-called Steiner tree. This is not true for gas network design problems. To see this, consider a network with two nodes u and v and two parallel gas pipes with $\beta_a = 1$. Assume the pressure is fixed to 3 at u and there is a flow demand of 2 from u to v. By symmetry the flow will be 1 one each of the two parallel arcs. Thus, $p_v = \sqrt{8}$, because $p_u^2 - p_v^2 = q_a|q_a| \Leftrightarrow 9 - p_v^2 = 1$. If we only had one arc, we would obtain $p_v = \sqrt{5}$. Thus, if there is a lower bound at v with $\sqrt{5} < \underline{p}_v \leq \sqrt{8}$, building two arcs is the only possibility.

2. One the other hand, adding arcs may also violate upper bounds. To see this, consider at network with three nodes s, t, and u and a demand of 2 from s to t. There are arcs (s,t) and (s,u) with $\beta_a = 1$. Assume that there is an upper bound $\overline{q}_{(s,u)} = 0.5$. Then adding an arc (u,t) will violate the flow bound on (s,u).

We use the networks `gaslib11`, `gaslib24`, `gaslib40`, and `gaslib134` from the Gaslib [54] and the `belgium` network of [15] as a building structure of our problem. To reformulate these already designed networks as new design problems, we replace each pipe with three parallel paths, where each path consists of a valve and a pipe placed sequentially; such parallel pipes are called loops in the gas literature. The valves ensure that the problem is monotone with respect to building additional pipes. The design variables in our problem, i.e, the variables x, are these newly introduced valves. We do not specify non-failable linking variables y.

With the additional paths, we introduced symmetry into the problem, therefore, we add for each group of new valves symmetry-breaking inequalities, which we also use in our algorithm according to Lemma 2.8.

In addition, we strenghten the formulation using two types of constraints. First, for each set of nodes $S \subset V$, which is separated from $\overline{S} \coloneqq V \setminus S$ by a bridge and $q(S) \coloneqq \sum_{v \in S} q_v \neq 0$, we add the *bridge* cut

$$\sum_{a \in \delta(S) \cap \tilde{A}^{\text{va}}} x_a \geq k + 1,$$

where $\delta(S) \coloneqq \delta^+(S) \cup \delta^-(S)$ and $\tilde{A}^{\text{va}} \subseteq A^{\text{va}}$ are all valves of the parallel paths for the pipes. Here, a bridge refers to one set of parallel paths whose removal turns the graph to be disconnected. These cuts are clearly valid.

The second type of inequalities stem from Gale's characterization of feasible b-flows, see Schrijver [55, Corollary 11.2f], and result in:

$$\sum_{a \in \delta^+(S)} \overline{q}_a \, x_a - \sum_{a \in \delta^-(S)} \underline{q}_a \, x_a \geq q(S).$$

If $\underline{q}_a \leq 0 \leq \overline{q}_a$, these cuts can be strengthened to

$$\sum_{a \in \delta^+(S)} \min\{\overline{q}_a, q(S)\} \, x_a + \sum_{a \in \delta^-(S)} \min\{-\underline{q}_a, q(S)\} \, x_a \geq q(S).$$

These constraints are generated for each (weakly) connected component S that arises, when removing all active elements (include the valves of potentially build arcs). All constraints are precomputed and added to the corresponding model up-front. In summary, we use Problem (20) for $\mathcal{X}$ together with the bridge cuts for the appropriate k and the above strengthened

Table 5: Gas network design test results for different settings and algorithms.

cut improvement strategy		time	# nodes	# solved
greedy	resolving			
–	–	93.58	5090.4	7
–	5	24.71	1006.7	8
–	∞	24.64	972.8	8
$\searrow$	–	34.31	1212.1	8
$\searrow$	5	25.34	986.0	8
$\searrow$	∞	25.10	973.1	8
$\nearrow$	–	29.68	1089.9	8
$\nearrow$	5	25.10	1040.4	8
$\nearrow$	∞	25.06	968.5	8
static SA		2090.54	40.10	2
dynamic SA		348.34	1067.5	5

Table 6: Detailed gas network test results using the resolving strategy.

k	network	time	time $\mathbb{F}$	time $\tilde{\mathcal{X}}$	# solves $\mathbb{F}$	# solves $\tilde{\mathcal{X}}$	$\lvert\mathbb{F}\rvert$	$\lvert\tilde{\mathcal{X}}\rvert$
1	`belgium`	16.01	3.81	1.68	614	74	5	71
1	`gaslib11`	4.03	0.20	0.15	19	17	3	19
1	`gaslib24`	3.90	0.51	0.40	39	24	4	41
1	`gaslib40`	3600.00	3592.14	3591.86	5	24	1	84
1	`gaslib134`	0.75	0.39	0.36	3	3	0	173
2	`belgium`	52.73	41.38	38.92	498	76	5	69
2	`gaslib11`	2.73	0.22	0.17	20	15	3	19
2	`gaslib24`	3.59	0.72	0.48	114	28	4	41
2	`gaslib40`	3600.00	3590.98	3590.71	8	34	2	91
2	`gaslib134`	13.03	12.50	12.28	3	14	0	173

cuts. The lower level constraints $\tilde{\mathcal{X}}$ are given by Problem (20) including the strengthened cuts and the bridge cuts for $k = 0$.

The computational results for our testset, see Table 5, are similar to the water network case. We can solve 8 out of 10 instances. The dynamic scenario approach performs better then the static approach. The later actually does not run for the larger networks `gaslib40` and `gaslib134` and $k = 2$, since it exceeds memory limits.

Table 6 presents detailed results for each network and number of failures k for the best performing setting, which uses resolving but no greedy strategy. We note that for `gaslib134` the bridge cuts are already sufficient to cut off non-resilient solutions, because this is a tree network. Indeed, our algorithm does not need to compute a solution in $\mathbb{F}$, i.e., no violated cut is generated. We are not able to solve `gaslib40`. One explanation is the long solving time of the separation problem over $\tilde{\mathcal{X}}$, which takes on average around 150 s per solve for $k = 1$.

We also solved the testset under the same settings without adding the symmetry breaking inequalities to $\mathbb{F}$ and $\tilde{\mathcal{X}}$. The shifted geometric mean of the solving times increased from about 25 s to about 166 s and the `belgium` network can not be solved anymore.

5 Conclusion

In this paper we developed a nested branch-and-cut method to solve a three-level model for optimization of fault-tolerant system design. Our approach is generic and allows to use general mixed-integer nonlinear formulations. Thus, many applications can be covered in principle. We have derived two classes of valid inequalities to describe the set of feasible integer solutions. Our implementation shows the validity of our approach on three examples. Clearly, the solved instances are relatively small so far. However, we stress that the applications are not-trivial, building on relatively complex and realistic models. Moreover, in light of the complexity result of Section 2.1, one important assumption is that k is small, which is sensible in most applications. This assumption does not, however, imply that the enumeration based methods of Section 1.3 are best. In fact, our method is almost always faster.

Future research directions include the application to further problems. One could experiment with more problem specific parameter settings and investigate primal heuristics to find good solution candidates. Our scheme furthermore allows to replace the lowest level problem (16) by problem specific algorithms, or to use heuristic solution algorithms to enhance the speed. Another direction is to search for further valid inequalities that allow to strengthen the formulations at the different levels.

Acknowledgements

We appreciate the helpful comments of Maximilian Gläser concerning the complexity result. The extensive computations would have not been possible without the implementations of Tristan Gally, Sonja Mars and Frederic Matter concerning SCIP-SDP and truss topology design as well as the work of Oliver Habeck for gas networks.

Funding

This research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – projects A4, A9 within SFB 805 (Project Number 57157498) and project A01 within CRC/Transregio 154 (Project Number 239904186).

References

[1] D. Adjiashvili, F. Hommelsheim, M. Mühlenthaler, and O. Schaudt. Fault-tolerant edge-disjoint paths – beyond uniform faults. *arXiv preprint 2009.05382*, 2020.

[2] D. Adjiashvili, S. Stiller, and R. Zenklusen. Bulk-robust combinatorial optimization. *Mathematical Programming*, 149(1):361–390, 2015.

[3] D. L. Alderson, G. G. Brown, W. M. Carlyle, and R. K. Wood. Solving defender-attacker-defender models for infrastructure defense. In R. K. Wood and R. F. Dell, editors, *Operations Research, Computing and Homeland Defense, INFORMS*, pages 28–49, Hanover, MD: Institute for Operations Research and the Management Sciences, 2011.

[4] N. Alguacil, A. Delgadillo, and J. M. Arroyo. A trilevel programming approach for electric grid defense planning. *Computers & Operations Research*, 41:282–290, 2014.

[5] L. C. Altherr, N. Brötz, I. Dietrich, T. Gally, F. Geßner, H. Kloberdanz, P. Leise, P. F. Pelz, P. Schlemmer, and A. Schmitt. Resilience in mechanical engineering – A concept for controlling uncertainty during design, production and usage phase of load-carrying structures. In P. F. Pelz and P. Groche, editors, *Uncertainty in Mechanical Engineering III*, volume 885 of *Applied Mechanics and Materials*, pages 187–198. Trans Tech Publications, 2018.

[6] L. C. Altherr, P. Leise, M. E. Pfetsch, and A. Schmitt. Resilient layout, design and operation of energy-efficient water distribution networks for high-rise buildings using MINLP. *Optimization and Engineering*, 20(2):605–645, 2019.

[7] S. Arora and B. Barak. *Computational complexity: A modern approach*. Cambridge University Press, 2009.

[8] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

[9] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[10] K. Bestuzheva, A. Gleixner, and S. Vigerske. A computational study of perspective cuts. *arXiv preprint 2103.09573*, 2021.

[11] D. Bienstock. *Electrical transmission system cascades and vulnerability: An operations research viewpoint.* SIAM-MOS Series on Optimization, 2015.

[12] C. Borraz-Sánchez, R. Bent, S. Backhaus, H. Hijazi, and P. V. Hentenryck. Convex relaxations for gas expansion planning. *INFORMS J. Comput.*, 28(4):645–656, 2016.

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

[14] R. L. Church and M. P. Scaparra. Protecting critical assets: The r-interdiction median problem with fortification. *Geographical Analysis*, 39(2):129–146, 2007.

[15] D. De Wolf and Y. Smeers. The gas transmission problem solved by an extension of the simplex algorithm. *Management Science*, 46(11):1454–1465, 2000.

[16] S. DeNegre. *Interdiction and discrete bilevel linear programming.* PhD thesis, Lehigh University, 2011.

[17] T. Ding, L. Yao, and F. Li. 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, 2018.

[18] C. D'Ambrosio, A. Lodi, S. Wiese, and C. Bragalli. Mathematical programming techniques in water network optimization. *European Journal of Operational Research*, 243(3):774–788, 2015.

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

[20] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

[21] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. Interdiction games and monotonicity, with application to knapsack problems. *INFORMS Journal on Computing*, 31(2):390–410, 2019.

[22] F. Furini, I. Ljubić, S. Martin, and P. San Segundo. The maximum clique interdiction problem. *European Journal of Operational Research*, 277(1):112–127, 2019.

[23] T. Gally. *Computational Mixed-Integer Semidefinite Programming.* PhD thesis, TU Darmstadt, 2019.

[24] T. Gally, A. Kuttich, M. E. Pfetsch, M. Schaeffner, and S. Ulbrich. Optimal placement of active bars for buckling control in truss structures under bar failures. In P. F. Pelz and P. Groche, editors, *Uncertainty in Mechanical Engineering III*, volume 885 of *Applied Mechanics and Materials*, pages 119–130. Trans Tech Publications, 2018.

[25] T. Gally, M. E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods and Software*, 33(3):594–632, 2018.

[26] G. Gamrath, D. Anderson, K. Bestuzheva, W.-K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S. J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M. E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig. The SCIP Optimization Suite 7.0. Technical report, Optimization Online, March 2020.

[27] N. Ghorbani-Renani, A. D. González, K. Barker, and N. Morshedlou. Protection-interdiction-restoration: Tri-level optimization for enhancing interdependent network resilience. *Reliability Engineering & System Safety*, 199:106907, 2020.

[28] A. M. Gleixner, H. Held, W. Huang, and S. Vigerske. Towards globally optimal operation of water supply networks. *Numerical Algebra, Control & Optimization*, 2(2):695–711, 2012.

[29] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. *Handbooks in operations research and management science*, 7:617–672, 1995.

[30] O. Habeck and M. E. Pfetsch. Combinatorial acyclicity models for potential-based flows. *Networks*, 2021. To appear.

[31] C. Hojny and M. E. Pfetsch. Polytopes associated with symmetry handling. *Mathematical Programming*, 175(1):197–240, 2019.

[32] J. Humpola. *Gas Network Optimization by MINLP.* PhD thesis, TU Berlin, 2014.

[33] J. Humpola and A. Fügenschuh. Convex reformulations for solving a nonlinear network design problem. *Comput. Optim. Appl.*, 62:717–759, 2015.

[34] E. Israeli. *System Interdiction and Defense.* PhD thesis, Naval Postgraduate School Monterey, 1999.

[35] E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[36] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[37] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. Preprint, Optimization Online, 2021.

[38] P.-M. Kleniati and C. S. Adjiman. A generalization of the Branch-and-Sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. *Computers & Chemical Engineering*, 72:373–386, 2015.

[39] T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe. *Evaluating gas network capacities.* MOS-SIAM Series on Optimization. SIAM, 2015.

[40] M. Kočvara. Truss topology design with integer variables made easy. Technical report, Optimization Online, 2010.

[41] L. Liberti. Symmetry in mathematical programming. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 263–283. Springer, 2012.

[42] L. Lozano and J. C. Smith. A backward sampling framework for interdiction problems with fortification. *INFORMS Journal on Computing*, 29(1):123–139, 2017.

[43] L. Lozano and J. C. Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.

[44] L. Lozano, J. C. Smith, and M. E. Kurz. Solving the traveling salesman problem with interdiction and fortification. *Operations Research Letters*, 45(3):210–216, 2017.

[45] F. Margot. Symmetry in integer linear programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 647–686. Springer, 2010.

[46] S. Mars. *Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design.* PhD thesis, FAU Erlangen-Nürnberg, 2013.

[47] A. Mitsos. Global solution of nonlinear mixed-integer bilevel programs. *Journal of Global Optimization*, 47(4):557–582, 2010.

[48] A. Morsi, B. Geißler, and A. Martin. Mixed integer optimization of water supply networks. In *Mathematical Optimization of Water Networks*, pages 35–54. Springer, 2012.

[49] MOSEK ApS. *MOSEK Optimizer API for C Release 9.2.35*, 2021.

[50] T. M. Müller, P. Leise, I.-S. Lorenz, L. C. Altherr, and P. F. Pelz. Optimization and validation of pumping system design and operation for water supply in high-rise buildings. *Optimization and Engineering*, pages 1–44, 2020. To appear.

[51] C. Papadimitriou. *Computational Complexity.* Addison Welsey, Reading, MA, 1994.

[52] M. E. Pfetsch, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, A. Morsi, J. Rövekamp, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. C. Steinbach, S. Vigerske, and B. M. Willert. Validation of nominations in gas network optimization: models, methods, and solutions. *Optimization Methods and Software*, 30(1):15–53, 2015.

[53] M. P. Scaparra and R. L. Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923, 2008.

[54] M. Schmidt, D. Aßmann, R. Burlacu, J. Humpola, I. Joormann, N. Kanelakis, T. Koch, D. Oucherif, M. E. Pfetsch, L. Schewe, R. Schwarz, and M. Sirvent. GasLib – A Library of Gas Network Instances. *Data*, 2(4):40, 2017.

[55] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[56] J. Schweiger and F. Liers. A decomposition approach for optimal gas network extension with a finite set of demand scenarios. *Optimization and Engineering*, 19:297–326, 2018.

[57] SCIP-SDP – a mixed integer semidefinite programming plugin for SCIP, 2019.

[58] T. C. Sharkey, S. G. Nurre Pinkley, D. A. Eisenberg, and D. L. Alderson. In search of network resilience: An optimization-based view. *Networks*, 77(2):225–254, 2021.

[59] J. C. Smith and C. Lim. Algorithms for network interdiction and fortification games. In A. Chinchuluun, P. M. Pardalos, A. Migdalas, and L. Pitsoulis, editors, *Pareto Optimality, Game Theory And Equilibria*, pages 609–644. Springer New York, New York, NY, 2008.

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

[61] H. v. Stackelberg. *Theory of the market economy.* Oxford University Press, 1952.

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

[63] K. Tanınmış and M. Sinnl. A branch-and-cut algorithm for submodular interdiction games. *arXiv preprint 2103.15788*, 2021.

[64] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[65] Q. Wang, J. Watson, and Y. Guan. Two-stage robust optimization for N-k contingency-constrained unit commitment. *IEEE Transactions on Power Systems*, 28(3):2366–2375, 2013.

[66] R. Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964.

[67] R. K. Wood. Bilevel network interdiction models: Formulations and solutions. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, 2011.

[68] Y. Xiang and L. Wang. An improved defender–attacker–defender model for transmission line defense considering offensive resource uncertainties. *IEEE Transactions on Smart Grid*, 10(3):2534–2546, 2019.

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

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

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

[72] K. Zheng and L. A. Albert. An exact algorithm for solving the bilevel facility interdiction and fortification problem. *Operations Research Letters*, 46(6):573–578, 2018.