EURO Journal on Computational Optimization 9 (2021) 100002

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

Robust flows with adaptive mitigation

Heiner Ackermann[a], Erik Diessel[a,*], Sven O. Krumke[b]

[a] Fraunhofer ITWM, Institute for Industrial Mathematics, Fraunhofer Platz 1, Kaiserslautern 67663, Germany
[b] Department of Mathematics, Technische Universität Kaiserslautern, Paul-Ehrlich-Str. 14, Kaiserslautern 67663, Germany

ARTICLE INFO

MSC:
90B06
90C11
90C47
90C35

Keywords:
Adjustable robust optimization
Supply chain
Cut-generation
Network flows

ABSTRACT

We consider an adjustable robust optimization problem arising in the area of supply chains: given sets of suppliers and demand nodes, we wish to find a flow that is robust with respect to failures of the suppliers. The objective is to determine a flow that minimizes the amount of shortage in the worst-case after an optimal mitigation has been performed. An optimal mitigation is an additional flow in the residual network that mitigates as much shortage at the demand sites as possible. For this problem we give a mathematical formulation, yielding a robust flow problem with three stages where the mitigation of the last stage can be chosen adaptively depending on the scenario. We show that already evaluating the robustness of a solution is NP-hard. For optimizing with respect to this NP-hard objective function, we compare three algorithms. Namely an algorithm based on iterative cut generation that solves medium-sized instances efficiently, a simple Outer Linearization Algorithm and a Scenario Enumeration algorithm. We illustrate the performance by numerical experiments. The results show that this instance of fully adjustable robust optimization problems can be solved exactly with a reasonable performance. We also describe possible extensions to the model and the algorithm.

1. Introduction

In supply chains, shortages caused by supply failures can have a significant impact, leading to large amount of lost sales and a possible reputation loss due to unfulfilled demand (Roberto and Levesque, 2004). Therefore it is important to hedge against these risks when making decisions affecting the supply chain configuration (Ho et al., 2015). Traditional optimization models consider only the associated risk of shortages (Hamdi et al., 2018) without taking into account possible reactions to failures. In practice, however, shortages can be mitigated by shipping additional material from unaffected suppliers, subject to capacity constraints. Ignoring this possibility leads to an overly pessimistic view of the risk and in suboptimal decisions with respect to realistic risk measures.

We use the following robustness measure to take these effects into account: the worst-case shortage still persisting at the the demand nodes *after* an optimal mitigation has been performed. An optimal mitigation is a flow in the residual network of the supplier capacities that mitigates as much shortage at the demand sites as possible. The goal is to choose a flow that is optimal with respect to this robustness measure.

Since the mitigation flow can depend on the failure scenario, we use the framework of adjustable robust optimization. The possible failures of suppliers are given by a set of scenarios, each corresponding to an event that leads to a complete production failure of one or more suppliers. In contrast to stochastic programming (Birge and Louveaux, 2011), we do not require any probability information for these scenarios. Instead we only consider the worst case.

We describe a mathematical formulation of the problem motivated above based on network flows. It can be considered as a tri-level robust flow problem. We give a linear programming (LP) formulation of the mitigation problem. Using this, we develop a robust problem that takes into account a mitigation that can be chosen adaptively on the scenario. The set of scenarios can be implicitly represented by the integral points of a chosen polyhedron. Thus, the set of scenarios can be chosen arbitrarily large and customized to the risk assessment.

We show that, in general, evaluating robustness for a given flow is NP-hard. We describe three algorithms that find an optimal flow with respect to the robustness objective.

The Outer Linearization Algorithm is based on a LP reformulation of the entire problem. The Scenario Enumeration Algorithm adds cuts for every scenario to an LP and re-solves it until convergence. The Iterative Cut Generation Algorithm iteratively finds the worst-case scenario for the current solution and adds a corresponding cut to the LP.

As the fastest algorithm, the Iterative Cut Generation Algorithm can solve small to medium-sized instances efficiently and performs better than the simpler alternative algorithms.

* Corresponding author.
E-mail addresses: heiner.ackermann@itwm.fraunhofer.de (H. Ackermann), erik.diessel@itwm.fraunhofer.de (E. Diessel), krumke@mathematik.uni-kl.de (S.O. Krumke).

https://doi.org/10.1016/j.ejco.2020.100002
Received 26 August 2020; Accepted 27 August 2020

1.1. Previous work

Robust Optimization deals with optimization problems where some of the coefficients, in the inequalities describing the feasible set, are uncertain. In contrast to stochastic programming, no probability distribution is assumed for these uncertain parameters. Instead, only a scenario set is given (in an implicit way) that describes the possible realizations of the uncertain parameters. The goal is to find a solution that is feasible with respect to all possible realizations of the parameters, i.e. the worst case is considered. Formally, a linear robust optimization problem can be written as

$$\min_{x}\{c^T x \mid A(\zeta)x \leq d \quad \text{for all } \zeta \in \mathcal{Z}\},$$

where $\mathcal{Z}$ represents the set of possible realizations of the uncertainty and $A(\zeta)$ is a matrix depending on the realized uncertainty ζ. We refer to the textbook by Ben-Tal et al. (2009) for a detailed treatment of the various types of Robust Optimization problems.

A more recent subfield of Robust Optimization is that of Adjustable Robust Optimization, introduced by Ben-Tal et al. (2004). It considers problems where a second-stage decision can be made that adjusts to the realization of the uncertain variables. The combination of first- and second-stage decision should however be feasible with respect to all possible realizations of the uncertain parameters. A recent survey was given by Yanikoğlu et al. (2018). The adjustable robust counterpart has the form

$$\min_{x,y(\cdot)}\{c^T x \mid A(\zeta)x + By(\zeta) \leq d \quad \text{for all } \zeta \in \mathcal{Z}\},$$

where x is the first-stage decision and $y(\zeta)$ is the second-stage decision that is a function of the realized uncertainty $\zeta \in \mathcal{Z}$.

In the literature, most algorithms deal with the case of second-stages decisions y that depend affinely on ζ (Affinely Adjustable Robust Optimization), i.e. there is a matrix Q and a vector y_0 such that $y(\zeta) = y_0 + Q\zeta$. This leads to easy decision rules for y (and often polynomially solvable problems), but the associated computed robustness might be too pessimistic. In particular, in our setting affine decision rules can be shown to be non-optimal, see Proposition 1. Results for the equivalence of adjustable and robust optimization in certain cases (Marandi and den Hertog, 2018), do not apply here. The optimal second-stage decision, i.e. the mitigation flow, depends in a highly nonlinear way on the failures of suppliers.

Bertsimas et al. (2013b) consider a similar problem as discussed in this paper. They describe a model for adaptive network flows. The goal is to find a nominal flow such that in case of arc failures the maximum flow value is maximized in the network where the arc capacities are given by the nominal flow values. The main difference to our model is, that we consider the amount of flow that is lost by supply failures and that our mitigation flow is bounded by the *residual* capacities in the network. Thus, both the structural properties of our problem and the corresponding algorithm are different. Also, our focus is more on obtaining exact optimal solutions instead of approximations.

Algorithms for solving Adjustable Robust Optimization problems have been proposed by multiple authors: Thiele et al. (2010) describe how a general Adjustable Robust Linear Optimization problem can be solved and give an IP-formulation for finding the worst-case scenario in the case of budgeted uncertainty. Zeng and Zhao (2013) extend the algorithm to also generate new columns and apply it to an adjustable robust location-transportation problem. Bertsimas et al. (2013a) applied this method to solve an adjustable robust problem in unit commitment of electric power systems. Simchi-Levi et al. (2019) describe an algorithm to solve Robust Network Flow problems based on cuts generated with an IP. Recently, Zhen et al. (2018) proposed to solve Adjustable Robust Optimization problems via Fourier-Motzkin elimination of the adjustable variables. However, due to the large number of adjustable variables in our problem, this approach would be inefficient in our case.

Several problems related to the design of a robust network have been studied in the literature. Alvarez-Miranda et al. (2014) discuss a recoverable version of the Two-Level Network Design problem. Garg and Smith (2008) consider a robust network design problem to ensure feasible multicommodity flows exists. Simchi-Levi et al. (2018) formulate a model for robust inventory optimization incorporating decisions on production quantities in the second stage. Cacchiani et al. (2016) as well as Atamtürk and Zhang (2007) and Ben-Tal et al. (2005) consider flow problems with demand uncertainties. The difference to our problem is that they do not consider shortages created by failures of suppliers.

The field of *Network interdiction* treats a subproblem of our problem: Given a network, the task is to find an interdiction strategy that reduces the performance of the network as much as possible. The interdiction strategy consists for example in removing a fixed number of edges from the network (Phillips, 1993; Wood, 1993). The problem of finding the worst-case scenario for a given flow in our robust setting can be interpreted as a network interdiction problem: Find an interdiction strategy (the scenario) such that the unmitigated shortage is maximized. The particular constraints and objective of the evaluation problem of finding the worst-case scenario in our setting are however not exactly equivalent to any of the classical network interdiction problems. Thus, we formulate a special MIP for solving this evaluation problem. One problem that is similar to our evaluation problem is the Maximum Residual Flow Problem considered by Aneja et al. (2001). In this problem, the goal is to find a flow such that the residual flow after an arc destruction is maximized. The problem is polynomially-time solvable for the case of single arc failures, but NP-hard in the case of 2 or more simultaneous arc failures (Du and Chandrasekaran, 2007).

In contrast to classical network interdiction where a bi-level problem occurs, we consider a tri-level optimization problem.

Under the notion of *defender-attacker-defender* models, tri-level network optimization problems are considered in areas related to infrastructure defense. An overview on this topic is given by Brown et al. (2006). Smith et al. (2007) discuss a three-stage problem where a designer constructs a network under some budget constraints, with an attacker then reducing the capacity of some arcs. The designer then chooses an optimal multi-commodity flow in the remaining network. They discuss several types of attackers that act either with a greedy or optimal strategy. They propose a cutting-plane algorithm based on linearization of a bilinear problem, employing similar techniques as in our Iterative Cut Generation Algorithm 2. Alderson et al. (2011) discuss a tri-level network model in the context of infrastructure defense. They use a decomposition algorithm based on mixed integer programs with quadratic constraints. Alderson et al. (2018) extend this model and analyze the results of a detailed case-study. The study of Brown et al. (2008) applies attacker-defender-attacker models to counter-terrorism efforts. Yuan et al. (2014) describe a defender-attacker-defender model for electrical power grids which they solve with a column-generation approach. Lozano and Smith (2017) discuss a backward sampling algorithm for solving tri-level network optimization problems with fortification. In the paper by Prince et al. (2013) a model for a three-stage procurement problem in supply chains is introduced, where an adversary can remove some of the capacities if no fortification is applied. Cappanera and Scaparra (2011) describe a tri-level model for optimizing the robustness of a network with respect to shortest-path distances and heuristics for an efficient solution.

1.2. Our contribution

In this paper, we provide a model for optimizing the robustness of a flow. Our robustness value of a flow corresponds to the negative of the worst-case shortage that can occur when taking this flow, after an optimal mitigation.

For this model, we show that affine decision rules do not lead to optimal robustness values. Instead, we consider arbitrary decision rules for the adaptive mitigation. We show that evaluating the robustness of a given solution is NP-hard. Nevertheless we are able to give an algorithm that yields an exact optimal solution with respect to this NP-hard objective function. For this problem, we compare three different algorithms, from a simple to a more advanced approach. The simplest is the Outer Linearization Algorithm that uses an explicit LP-reformulation of the robustness. The Scenario Enumeration Algorithm maintains an approximation of the robustness value as an LP. In each iteration, this LP is re-solved and for each scenario a cut for the current solution is added, until the evaluated robustness matches a bound. The Iterative Cut Generation Algorithm also maintains an outer LP. In each iteration, the worst-case scenario for the current solution is found by solving a MIP. A corresponding cut is added to the LP which is then re-solved. This is repeated until the evaluated robustness of the current solution matches the bound given by the optimal LP objective value.

Experiments show that the fastest Iterative Cut Generation Algorithm allows to solve medium-sized instances efficiently. We also describe how the algorithm can be adapted for solving various extensions of the model.

This paper is organized as follows: In Section 2 we formally define the basic model. In Section 3 we show that evaluating a given flow for its robustness in the adaptive setting is NP-hard. Section 4 presents our algorithms, the corresponding experimental results are given in Section 5. We conclude in Section 6 where we also discuss extensions of the basic model.

2. Our model

We consider a complete bipartite network $G = (V = S \cup D, E)$ of suppliers S and demand nodes D, where a single commodity of material is distributed. Each supplier $s \in S$ has a maximal production capacity $C_s \geq 0$ and the arc $(s, d) \in E$ between s and d has capacity $c_{s,d} \geq 0$ limiting the amount of material that can be simultaneously shipped over the arc. For notational simplicity we assume that the network is a complete bipartite graph, since a non-existing arc can be represented by one with zero capacity. Each demand node $d \in D$ has an associated demand $b_d \geq 0$.

We call the flow that is yet unaffected by a failure the *nominal flow*. Our goal is to choose a nominal flow that performs best with respect to our robustness measure. A nominal flow is feasible if the following *nominal constraints* are fulfilled:

$$\sum_{s \in S} f_{s,d} = b_d \qquad \text{for all } d \in D \tag{1a}$$

$$\sum_{d \in D} f_{s,d} \leq C_s \qquad \text{for all } s \in S \tag{1b}$$

$$0 \leq f_{s,d} \leq c_{s,d} \qquad \text{for all } (s,d) \in E \tag{1c}$$

Subsequently we refer to the conditions in (1) as the nominal constraints. The failure scenario is given as a vector x with a component $x_s \in \{0, 1\}$ for each supplier $s \in S$. The entry x_s is 1 when supplier s has a failure and 0 otherwise. This means that a flow of $x_s f_{s,d}$ is lost between supplier s and demand node $d \in D$. The mitigation flow $f'_{s,d}$ can be chosen after the realization of the scenario. We assume that the nominal flow f cannot be adapted to the scenario. In practice, this restriction is due to the fact that already agreed on shipments often cannot be changed anymore. Procurement contracts are often binding for relatively long time frames and produced material is in many cases specifically customized to the destination. Hence, for the mitigation, only more flow can be sent, subject to the residual capacities. We assume that the capacity for mitigation is bounded by the same constraints as the nominal flow, i.e. the sum of nominal and mitigation flow must be below the capacity limit for each arc and each supplier.

The mitigation problem can now be written as:

Definition 1 (Mitigation problem)**.**

$$z^*_N(x, f) := \max \sum_{s \in S} \sum_{d \in D} f'_{s,d} \tag{2a}$$

$$\text{s.t.} \qquad f'_{s,d} \leq (c_{s,d} - f_{s,d})(1 - x_s) \quad \text{for all } s \in S \text{ and } d \in D \tag{2b}$$

$$\sum_{d \in D} f'_{s,d} \leq \left(C_s - \sum_{d \in D} f_{s,d} \right)(1 - x_s) \quad \text{for all } s \in S \tag{2c}$$

$$\sum_{s \in S} f'_{s,d} \leq \sum_{s \in S} x_s \cdot f_{s,d} \qquad \text{for all } d \in D \tag{2d}$$

$$f'_{s,d} \geq 0 \qquad \text{for all } s \in S \text{ and } d \in D \tag{2e}$$

The constraints (2b) ensure that the mitigation flow over the arc (s,d) is 0 whenever the supplier s has a failure, i.e. $x_s = 1$. Otherwise, the mitigation is bounded by the residual capacity $c_{s,d} - f_{s,d}$. Similarly, (2c) ensures that the mitigation flow does not exceed the residual capacity of the supplier. Failures do not need to be considered here because of (2b). Constraint (2d) ensures that at each demand node the amount of mitigation is not higher than the shortage. Since the failure x and the nominal flow f are given explicitly in the mitigation problem, it corresponds to a capacitated transportation problem which can be solved in polynomial time (Ahuja et al., 2014).

Remark 1. The formulation of the linear program in Definition 1 has a large influence on the performance of the later algorithms. Since the constraint (2b) already implies $f'_{s,d} = 0$ for every $s \in S$ with $x_s = 1$, the constraint (2c) could be reformulated as

$$\sum_{d \in D} f'_{s,d} \leq C_s - \sum_{d \in D} f_{s,d}$$

without changing the set of feasible solutions to this linear program. However, the later reformulations in our algorithms would be affected. It would lead to cuts that are less restricting and thus to longer running times. In particular, the running time of the Iterative Cut Generation Algorithm described later would increase by a factor of up to 10.

The set of scenarios will be denoted by X. Each scenario $x = (x_s)_{s \in S} \in X$ gives the status of every supplier. In this way we can consider scenarios with simultaneous failures of multiple suppliers. The unmitigated shortage in scenario x with the mitigation flow f' is given by the difference of the mitigated amount $z^*_N(x, f)$ and the total shortage $\Sigma_{s \in S} \Sigma_{d \in D} x_s f_{s,d}$. We now have all prerequisites for defining the main subject of our paper.

Definition 2 (Robust Flow problem with adaptive mitigation)**.** The Robust Flow problem with adaptive mitigation asks to optimize for a given capacitated network and set of scenarios X the unmitigated shortage:

$$\begin{aligned} &\max_f \ \min_{x \in X} \left(z^*_N(x, f) - \textstyle\sum_{s \in S} \sum_{d \in D} x_s f_{s,d} \right) \\ &\text{s.t. nominal flow constraints (1) for } f \end{aligned} \tag{3}$$

We call the objective value in (3) obtained by a nominal flow f its *robustness value*. To treat scenario sets which are too large to write down explicitly, we represent the scenario sets implicitly as the integral points in a polyhedron.

Definition 3 (Implicit representation of a scenario set)**.** An implicit representation of a scenario set is given by a weight matrix $A \in \mathbb{R}^{k \times |S|}$ and a parameter vector $\Gamma \in \mathbb{R}^k$, where $k \in \mathbb{N}$. The corresponding scenario set is then given by:

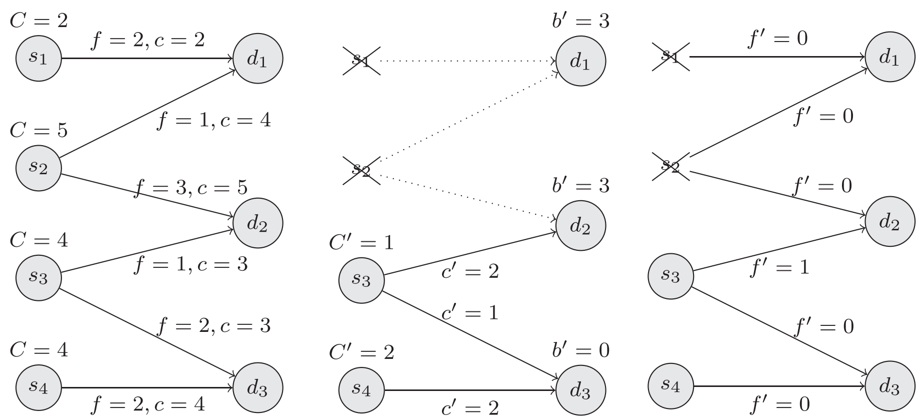
$$X(A, \Gamma) := \left\{ x \in \{0,1\}^{|S|} \;\middle|\; Ax \leq \Gamma \right\}.$$

We consider in particular the Γ-robust scenario set consisting of all subsets of suppliers of size at most Γ

$$X(\Gamma) := \left\{ x \in \{0,1\}^{|S|} \;\middle|\; \sum_{s \in S} x_s \leq \Gamma \right\}.$$

Example 1 (Illustrative solution of an instance)**.** In Fig. 1, an example instance is shown together with an optimal mitigation.

Non-optimality of affine decision rules. A decision rule in the context of adjustable robust optimization is a mapping from the scenario variables to second-stage decisions. In our case, this would correspond to a



(a) The nominal flow f and the supplier capacities C and arc capacities c.

(b) The residual capacities C' of the suppliers, c' of the arcs and the shortages b' at the demand nodes, in the scenario of a failure of suppliers s_1 and s_2.

(c) An optimal mitigation that mitigates a total of 1 unit of flow out of a shortage of 6 units.

Fig. 1. Example instance of the Robust Flow problem.

mapping from the scenario set X to mitigation flows f'. In the literature, Adjustable Robust Optimization problems are often tackled by restricting the decision rules to affine functions in the scenario variables to achieve an approximation (Ben-Tal et al., 2004).

The following proposition shows that the restriction to affine decision rules significantly reduces the obtained robustness value in our model and is therefore not suitable as an approximation.

Proposition 1 (Non-optimality of affine decision rules). *There exists an instance of the Robust Flow problem where the worst-case unmitigated shortage with fully adjustable mitigation is 0 whereas the restriction to an affine rule for the mitigation leads to an unmitigated shortage of 1. In particular, the ratio between the two values and the additive gap can be arbitrarily large.*

A proof of Proposition 1 can be found in the appendix.

3. Hardness

In this section, we show that computing the robustness value, i.e. finding the worst-case scenario, is NP-hard, when the scenario set is given as Γ-robust scenario set in an implicit representation.

Definition 4 (Evaluation Problem). An instance of the Flow Robustness Evaluation Problem consists of a network and a parameter Γ. For the given network and nominal flow f, the goal is to compute

$$\min_{x\in X(\Gamma)}\left(z^*_N(x,f)-\sum_{s\in S}\sum_{d\in D}x_s f_{s,d}\right).$$

For this problem we will show NP-hardness.

Theorem 1. (NP-*hardness of the Evaluation Problem)* *The Flow Robustness Evaluation Problem is strongly* NP*-hard, even if there are only arc capacities and all supplier capacities are unbounded.*

For proving the NP-hardness, we first show that a certain nonlinear maximization problem can be represented as an evaluation problem. We then give a reduction from the clique problem to this nonlinear maximization problem. Using the representation, this can in turn be reduced to the Evaluation Problem.

We assume that the capacity of each supplier is unlimited, only the edges have capacities. The amount of shortage at a demand node d is given by

$$\sum_{s\in S} x_s\cdot f_{s,d}.$$

The capacity which can be used for mitigation at demand node d is then given by

$$\sum_{s\in S}(c_{s,d}-f_{s,d})(1-x_s).$$

Since the suppliers have unlimited capacity, we can always send the maximal possible amount of mitigation allowed by the arc capacity. This does not interfere with the mitigation for other demand nodes. Hence each demand node can be considered separately. Note that we cannot mitigate more than the shortage. Hence the unmitigated amount at the demand node d is given by

$$\left(\sum_{s\in S} x_s\cdot f_{s,d}-\sum_{s\in S}(c_{s,d}-f_{s,d})(1-x_s)\right)^+$$
$$=\left(\underbrace{\sum_{s\in S} f_{s,d}}_{=b_d=\text{const.}}-\underbrace{\sum_{s\in S} c_{s,d}}_{=\text{const.}}+\sum_{s\in S} c_{s,d}x_s\right)^+,$$

where the expression $(g)^+ := \max\{0, g\}$ denotes the nonnegative part of g.

Since the sums $\sum_{s\in S} f_{s,d} = b_d$ and $\Sigma_{s\in S}c_{s,d}$ are constant and independent of f, the unmitigated amount in this case is *independent* of f. If we assume that the flow is feasible, we also have $f_{s,d} \le c_{s,d}$. Thus the expression above can be rewritten as

$$\left(\left(\sum_{s\in S} c_{s,d}x_s\right)-a_d\right)^+,$$

where $a_d := \sum_{s\in S} c_{s,d} - b_d \ge 0$ is the constant corresponding to the demand b_d and capacities $c_{s,d}$ for the considered demand node $d \in D$. Note that each possible value of $0 \le a_d \le \Sigma_{s\in S}c_{s,d}$ can be obtained by choosing appropriate values for the demand b_d. By summing up over all demand nodes $d \in D$, we obtain the total unmitigated shortage, i.e. the negative of the objective function in Definition 4

$$\sum_{d\in D}\left(\left(\sum_{s\in S} c_{s,d}x_s\right)-a_d\right)^+=-\left(z^*_N(x,f)-\sum_{s\in S}\sum_{d\in D}x_s f_{s,d}\right).$$

Hence, because maximization of the negative objective is equivalent to minimization of the original objective, the question of finding a worst-case scenario according to Definition 4 is equivalent to solving the maximization problem

$$\max_{x\in X(\Gamma)}\sum_{d\in D}\left(\left(\sum_{s\in S} c_{s,d}x_s\right)-a_d\right)^+. \tag{4}$$

With these considerations, we are now able to give a reduction from the clique problem to the Evaluation Problem, yielding a proof for Theorem 1.

Algorithm 1: Scenario Enumeration

Initialize $(*)$ to be the following LP:
$\max t$
s.t. nominal flow constraints (1) for f
$t \leq 0$
repeat
  Solve $(*)$, let f^* be the corresponding optimal solution and let t^* be its value
  foreach *scenario* x **do**
    Find a maximal $s_0 - d_0$ flow in the network for the mitigation in scenario x with the current f^* according to Definition 5; let $v^*(x)$ be its value
    Let (U, W) be the corresponding minimum cut
    set for $d \in D$, $s \in S$: $y_d := \begin{cases} 1, d \in U, \\ 0, d \in W, \end{cases}$

$$\xi_s := \begin{cases} 0, s \in U, \\ 1, s \in W \end{cases} \quad \beta_{s,d} := \max\{1 - \xi_s - y_d, 0\}$$

    Add the constraint

$$t \leq \sum_{s\in S}\sum_{d\in D}(c_{s,d} - f_{s,d})(1 - x_s)\beta_{s,d} + \sum_{s\in S}\left(C_s - \sum_{d\in D} f_{s,d}\right)(1 - x_s)\xi_s + \sum_{s\in S}\sum_{d\in D} f_{s,d} x_s y_d - \sum_{s\in S}\sum_{d\in D} f_{s,d} x_s$$

    to $(*)$
  end

$$\rho^* := \min_{\text{scenario } x} v^*(x) - \sum_{s\in S}\sum_{d\in D} f_{s,d} x_s$$

until $\rho^* = t^*$
return last f^*

Proof of Theorem 1. We use a reduction from the well-known NP-hard clique problem. Given is an undirected graph $G = (V, E)$. The question is, whether there exists a clique of size k in G. We claim that this can be modeled, setting $\Gamma := k$ as

$$\max_{0\leq x\leq 1:\sum_{v\in V} x_v \leq \Gamma} \sum_{v\in V}\left(3\Gamma \cdot x_v - (4\Gamma - 2) + \sum_{u:(v,u)\in E} x_u\right)^+.$$

There exist corresponding values for the capacities c and demands b modeling the above problem in the form of (4), since we can assume w.l.o.g. that $\deg(v) \geq \Gamma - 2$ holds for all nodes $v \in V$, as otherwise the node v could not be part of any clique of size $\Gamma = k$, leading to a reduction by removing this node.

We now show that the solution of the above problem has objective value $\geq \Gamma$ if and only if there is a clique of size Γ. Suppose that $C \subseteq V$ is a clique of size Γ in G. Then, setting

$$x_v := \begin{cases} 1, & v \in C, \\ 0, & v \notin C, \end{cases}$$

yields a solution value of Γ, since we then have for every $v \in C$, i.e. $|C| = \Gamma$ times:

$$3\Gamma \cdot x_v - (4\Gamma - 2) + \sum_{u:(v,u)\in E} x_u = 3\Gamma - (4\Gamma - 2) + \Gamma - 1 = 1.$$

Suppose x is a solution to the maximization problem with objective value at least Γ. Since $\Sigma_{v \in V} x_v \leq \Gamma$ holds by assumption, a requirement for

$$3\Gamma \cdot x_v - (4\Gamma - 2) + \underbrace{\sum_{u:(v,u)\in E} x_u}_{\leq \Gamma} \overset{?}{>} 0$$

being positive is that $3\Gamma \cdot x_v > 3\Gamma - 2$, i.e. $x_v > \frac{3\Gamma-2}{3\Gamma}$. We set $C := \left\{ v \in V \mid x_v > \frac{3\Gamma-2}{3\Gamma} \right\}$ and show that C is a clique of size Γ. Since

$$\Gamma \geq \sum_{v\in V} x_v > |C| \cdot \frac{3\Gamma - 2}{3\Gamma},$$

we get from the assumption $\Sigma_{v \in V} x_v \leq \Gamma$ the bound

$$|C| < \Gamma \cdot \frac{3\Gamma}{3\Gamma - 2} = \Gamma + \frac{2}{3} + \frac{4}{9\Gamma - 6} < \Gamma + 1,$$

where we assumed w.l.o.g. that $\Gamma \geq 3$. Since $|C|, \Gamma \in \mathbb{N}$, this implies $|C| \leq \Gamma$. Every individual term is bounded in the following way:

$$3\Gamma \cdot x_v + \sum_{u:(v,u)\in E} x_u - (4\Gamma - 2) \leq 3\Gamma + \Gamma - 1 - (4\Gamma - 2) = 1.$$

Because of our previous considerations there are at most $|C| \leq \Gamma$ many positive terms and the objective value is at least Γ, this implies that there are exactly Γ terms with value 1 (belonging to the nodes $v \in C$) and $|C| = \Gamma$.

We now show that C is a clique. Assume the contrary, i.e. there are two nodes v, $w \in C$ such that $(v, w) \notin E$. By the definition of C, $x_w > \frac{3\Gamma-2}{3\Gamma} > 0$. Thus, the term corresponding to x_v has the following upper bound

$$3\Gamma \cdot x_v - (4\Gamma - 2) + \sum_{u:(v,u)\in E} x_u \leq 3\Gamma - (4\Gamma - 2) + \Gamma - 1 - x_w \leq 1 - x_w < 1,$$

contradicting our conclusion above that the term has value 1.

Note that the transformation of an instance of the clique problem to the corresponding instance of the maximization problem and its transformation to an evaluation problem can be done in polynomial time. Also, all used numerical data in the maximization problem is integral and bounded linearly in the size of the clique instance. Thus, we can also encode the data in unary with only a linear overhead. This implies that the problem is *strongly* NP-hard. □

Corollary 1 (NP-hardness of Robust flow problem)**.** *The Robust Flow Problem with adaptive mitigation (Definition 2) is* NP*-hard.*

Proof. The instances we used in the proof of the NP-hardness of the Evaluation Problem have the property that their evaluation value is independent of the nominal flow f. Thus, computing the evaluation value for these instances is equivalent to computing the (optimal) objective value for the corresponding robust flow problem. Hence, the decision problem version of the optimization problem for a given bound $t \in \mathbb{R}$

"Is the optimal objective value of the Robust Flow instance at most t?"

is at least as hard as the evaluation problem, i.e. NP-hard. The Robust Flow problem is thus also NP-hard. □

It is well-known that the clique problem is W[1]-complete (Downey and Fellows, 1999). Thus, since the parameter Γ is carried over in the above construction, the previous theorem in fact implies that the Evaluation Problem is also W[1]-complete and, therefore, most likely not fixed-parameter tractable with respect to the parameter Γ.

4. Algorithms

In this section, we describe three algorithms for the Robust Flow problem with adaptive mitigation, in increasing order of sophistication.

4.1. Outer Linearization Algorithm

We start with presenting the conceptually most simple algorithm for solving an Adjustable Robust Optimization problem. The *Outer Linearization Algorithm* is based on including the mitigation for all scenarios as variables in a LP. The objective function is represented as the maximum of the unmitigated shortages over each scenario.

$$
\begin{array}{lr}
\max\ t & \\
\text{s.t. nominal flow constraints (1) for } f & \\
\quad t \le \sum_{s\in S}\sum_{d\in D} f'^{(x)}_{s,d} - \sum_{s\in S}\sum_{d\in D} x_s f_{s,d} & \text{for all} x \in X \\
\quad f'^{(x)}_{s,d} \le (c_{s,d} - f_{s,d})(1-x_s) & \text{for all } s\in S, d\in D \text{ and } x\in X \\
\quad \sum_{d\in D} f'^{(x)}_{s,d} \le \left(C_s - \sum_{d\in D} f_{s,d}\right)(1-x_s) & \text{for all } s\in S \text{ and } x\in X \\
\quad \sum_{s\in S} f'^{(x)}_{s,d} \le \sum_{s\in S} x_s \cdot f_{s,d} & \text{for all} d\in D \text{and} x\in X \\
\quad f'^{(x)}_{s,d} \ge 0 & \text{for all } s\in S, d\in D \text{ and } x\in X
\end{array} \tag{5}
$$

The formulation as one large LP is however very memory-intensive in practice, since the number of mitigation variables $|X|\cdot|S|\cdot|D|$ might become too large. In situations where the number of scenarios is small, this approach however is usually fast enough for practical purposes.

4.2. Scenario Enumeration Algorithm

In this section, we describe the Scenario Enumeration Algorithm. Instead of evaluating the robustness of a given flow using an IP, we will use an enumeration of the scenarios and solve for each scenario the mitigation problem by computing a maximum flow. The corresponding minimum cut is then used to deduce a dual upper bound for the robustness. The cut is then inserted into the Master-LP and the LP is re-solved. We iterate this, until the upper bound equals the current value of the robustness.

Definition 5 (Maximum flow network)**.** For a flow *f*, the maximum flow network for the mitigation in failure scenario *x* is given by the following parts:

- the nodes representing all suppliers S and all demand nodes D
- a supersource s_0, with arcs (s_0, s), $s\in S$ each having capacity

$$c'_{s_0,s} = \left(C_s - \sum_{d\in D} f_{s,d}\right)(1-x_s)$$

- arcs from every supplier $s\in S$ to every demand node $d\in D$ with capacity $c'_{s,d} = (1-x_s)(c_{s,d} - f_{s,d})$
- a supersink d_0 with arcs (d, d_0), $d\in D$, each having capacity $c'_{d,d_0} = \sum_{s\in S} x_s f_{s,d}$

Theorem 2 (Correctness of Alg. 1)**.** *Let f^* be the output of Algorithm 1. Then, there does not exist a feasible flow f which is more robust than f^*.*

Proof. The constraints that are added to (*) correspond to the capacity of a cut in the maximum flow network of Definition 5. The capacity of this cut provides an upper bound to the total amount of flow in the case of scenario *x*. Since we can formulate the robustness maximization of the flow as the problem to maximize the amount of flow that can arrive in total in the worst-case scenario, this is equivalent to an upper bound on the robustness value of the flow.

Since these constraints on *t* represent all valid upper bounds, the solution t^* that is obtained always overestimates the robustness value of an optimal solution. The value ρ^* gives the robustness value of the current flow *f*. Thus, always $\rho^* \le t^*$ holds. If $\rho^* = t^*$ then the current solution has a robustness value equal to the upper bound t^*, which implies that it must be optimal. □

4.3. Cut-generation algorithms for the robust flow problem

We now describe how a cut generation algorithm can solve the Robust Flow problem. The main difference to the other presented algorithms is that we allow here an implicitly represented set of scenarios $X(A,\Gamma)$ according to Definition 3. This makes the other algorithms impractical, since they require to enumerate all scenarios. Random sampling from the scenario set is also not sufficient, since we want an exact solution. Thus we use a cutting plane approach.

We use techniques originating from Benders' decomposition (Benders, 1962) and linearization with McCormick envelopes (McCormick, 1976). This general technique has also been applied to several tri-level optimization problems, e.g. by Alderson et al. (2011); Sadeghi et al. (2017); Smith et al. (2007), although the specific formulations vary with different models.

Algorithm 2: Iterative Cut Generation Algorithm

Initialize (∗) to be the following LP:
$\max t$
s.t. nominal flow constraints (1) for f
$t \le 0$
repeat
  Solve (∗), let f^* be the corresponding optimal solution with value t^*
  Solve (7) with the current f^*, let (z^*, μ^*, π^*, x^*) be the corresponding optimal solution with objective value ρ^*
  Add the constraint

$$t \le \sum_{s\in S}\sum_{d\in D}(c_{s,d} - f_{s,d}) z^*_{s,d} + \sum_{s\in S}\left(C_s - \sum_{d\in D} f_{s,d}\right)\mu^*_s + \sum_{s\in S}\sum_{d\in D} f_{s,d}\pi^*_{s,d} - \sum_{s\in S}\sum_{d\in D} f_{s,d} x^*_s$$

  to (∗)
until $\rho^* = t^*$;
return last f^*

We start with dualizing the model for the mitigation flow (2). Given some nominal flow *f* and scenario *x*, this results in the following linear program.

$$
\begin{array}{lll}
\min & \sum_{s\in S}\sum_{d\in D}(c_{s,d} - f_{s,d})(1-x_s)\beta_{s,d} & \\
& + \sum_{s\in S}\left(C_s - \sum_{d\in D} f_{s,d}\right)(1-x_s)\xi_s + \sum_{d\in D}\sum_{s\in S} f_{s,d} x_s y_d & \\
\text{s.t.} & \xi_s + \beta_{s,d} + y_d \ge 1 & \text{for all } s\in S \text{ and } d\in D \\
& \xi_s, \beta_{s,d}, y_d \ge 0 & \text{for all } s\in S \text{ and } d\in D
\end{array} \tag{6}
$$

By strong duality we have that the optimal objective values of the primal and the dual coincide for every fixed value of *f* and *x*. Thus we can use the dual to find a failure scenario *x* such that the unmitigated amount is maximal.

We first have to linearize the terms $(1-x_s)\beta_{s,d}$, $(1-x_s)\xi_s$ and $x_s y_d$ that appear in the objective function. Since we require that $x_s \in \{0,1\}$, we can replace $x_s y_d$ by the variable $\pi_{s,d}$ with the constraints $\pi_{s,d} \ge x_s + y_d - 1$ and $\pi_{s,d} \ge 0$. Since the coefficient in front of $x_s y_d$ is always positive, in an optimal solution we will choose $\pi_{s,d} = \max\{x_s + y_d - 1, 0\}$. Since $x_s \in \{0,1\}$ and $y_d \in [0,1]$, we then have

$$\pi_{s,d} = \begin{cases} 0, & x_s = 0, \\ y_d, & x_s = 1, \end{cases} = x_s \cdot y_d.$$

Similarly, we can replace $(1-x_s)\beta_{s,d}$ with the variable $z_{s,d}$ and the constraints $z_{s,d} \ge \beta_{s,d} - x_s$ and $z_{s,d} \ge 0$. Analogously, the term $(1-x_s)\xi_s$ can be replaced with the variable μ_s and the constraints $\mu_s \ge \xi_s - x_s$ and $\mu_s \ge 0$.

This leads to the following IP:

$$
\begin{aligned}
\min & \textstyle\sum_{s\in S}\sum_{d\in D}(c_{s,d}-f_{s,d})z_{s,d}+\sum_{s\in S}\left(C_s-\sum_{d\in D}f_{s,d}\right)\mu_s \\
&+\textstyle\sum_{s\in S}\sum_{d\in D}f_{s,d}\pi_{s,d}-\sum_{s\in S}\sum_{d\in D}f_{s,d}x_s \\
\text{s.t.}\quad & \xi_s+\beta_{s,d}+y_d\ge 1 && \text{for all } s\in S \text{ and } d\in D \\
& Ax\le\Gamma \\
& z_{s,d}\ge\beta_{s,d}-x_s && \text{for all } s\in S \text{ and } d\in D \\
& \mu_s\ge\xi_s-x_s && \text{for all } s\in S \\
& \pi_{s,d}\ge x_s+y_d-1 && \text{for all } s\in S \text{ and } d\in D \\
& \xi_s,\beta_{s,d},y_d,z_{s,d},\mu_s,\pi_{s,d}\ge 0 && \text{for all } s\in S \text{ and } d\in D \\
& x_s\in\{0,1\} && \text{for all } s\in S
\end{aligned}
\tag{7}
$$

The idea of our algorithm is based on evaluating (7) for the current flow f to get an exact value of the robustness of f and to generate a corresponding linear bound for the robustness of an arbitrary flow using the optimal solution of (7).

Theorem 3 (Correctness of Algorithm 2)**.** *Let f^* be the output of Algorithm 2. Then, there does not exist a feasible flow f which is more robust than f^*.*

Proof. Let (z^*, μ^*, π^*, x^*) be a feasible solution to (7). Note that feasibility is independent of the flow f. Thus, since (7) is a minimization problem, the term

$$
\sum_{s\in S}\sum_{d\in D}(c_{s,d}-f_{s,d})z^*_{s,d}+\sum_{s\in S}\left(C_s-\sum_{d\in D}f_{s,d}\right)\mu^*_s
+\sum_{s\in S}\sum_{d\in D}f_{s,d}\pi^*_{s,d}-\sum_{s\in S}\sum_{d\in D}f_{s,d}x^*_s
$$

is an upper bound on the optimal objective value of (7). Since we add to (*) only constraints of this form (for feasible (z^*, μ^*, π^*, x^*)), the objective value t^* of (*) in the last iteration is an upper bound on the optimal robustness of any feasible flow. Algorithm 2 terminates only when $\rho^*\ge t^*$. Thus, f^* has a robustness value of ρ^* which is at least as large as the upper bound for the robustness of any feasible flow. Hence, f^* has optimal robustness. □

The finite convergence of Algorithm 2 can be seen as follows: Each solution of the IP (7) corresponds to a cut in the network. The robustness value is completely characterized by the capacity of these cuts. Thus, at the latest after all possible cuts have been added, the algorithm terminates.

The advantage of the Scenario Enumeration Algorithm 1 compared to the Iterative Cut Generation Algorithm 2 is its flexibility: Scenarios can be chosen arbitrarily. Also, with a slight modification we can assign probabilities to the scenarios to model not only worst-case robustness but also probabilistic robustness in the form of: The unmitigated value with 95% probability is at least a given value. This allows to optimize with respect to alternative risk measures, like the conditional value-at-risk, which are heavily used in financial mathematics (Rockafellar and Uryasev, 2000).

5. Numerical results

We implemented the algorithms presented above in Python 3.7. For solving the LPs and IPs we used Gurobi version 8.1 (Gurobi Optimization, LLC, 2019). The max-flow problems in the Scenario Enumeration algorithm are solved with NetworkX (Hagberg et al., 2008). The benchmarks were run on a personal laptop running Windows 10 with an Intel i7-8665U CPU with 1.9 GHz and 16 GB of RAM.

We refer to the master thesis of one of the authors (Diessel, 2019), on which this paper is based, for additional numerical studies.

We generate instances of our Robust Flow problem with NETGEN (Klingman et al., 1974) which is widely used throughout the literature to evaluate network flow algorithms. We are using an implementation by Schlenker (1989). Table 1 illustrates the chosen parameter values for NETGEN. This ensures that we get bipartite graphs.

NETGEN generates capacitated network flow problems. Since the generated total capacity of the source nodes is equal to the total demand, no residual capacity is available in the generated networks. Thus, we multiply each source capacity by a common parameter $\eta > 1$ to yield some residual capacity for mitigation. We used the value $\eta = 1.1$ in all the experiments, unless otherwise noted. For each set of parameters we created 100 instances using different random seeds and measured the average running time on them.

By using a model including adaptive mitigations, we can improve significantly on the robustness value. This is illustrated in Fig. 2, showing the different resulting objective values of an optimal solution obtained in our model including mitigation, without mitigation and a solution where we considered mitigation for the first found flow (which we call the *naive solution*).

In all figures, we show the objective value as a positive quantity, i.e. with reversed sign in comparison to the original formulation (3), for a better overview. This means that lower values are better and correspond to more robust solutions with less shortages. The parameter Γ of failing suppliers varies.

In the classical robust model without mitigations, the mitigations are forced to be 0. Thus, the classical robustness value is given by the sum of shortages without being improved by mitigations. The plot shows that taking mitigations into account provides a benefit in the robustness value, particularly if the parameter Γ is not too big, i.e. $\Gamma\le 3$. For larger values of Γ the scenarios become so restricting, that also mitigation cannot improve the shortage significantly. Additionally, we show the adjustable robust objective value that would be obtained by a naive optimization (i.e. by just choosing the first found feasible flow and only considering the mitigation afterwards). For small values of Γ, this naive solution is significantly worse than the optimal one. For larger values of Γ however there is no difference anymore, as the scenarios become so difficult that the choice of a flow does not impact the robustness value.

In Fig. 3, we show the influence of the parameter η controlling the capacity of the suppliers on the obtained objective value for a network instance. As can be seen, for small values of $\eta < 1.1$, there is no big difference between the classical robust optimization and the inclusion of mitigation. This is due to the fact that the residual capacities are so small that no effective mitigation can be provided. At larger capacities ($\eta\ge 1.2$), the difference between the two optimization models becomes larger. Larger over-capacities than $\eta = 1.2$ do not benefit the mitigations anymore, since the network structure is then the bottleneck instead of the capacity values. This plot also shows why we have chosen the value $\eta = 1.1$ for the other experiments: it provides a value where the first differences between the different optimization models arises, without having too extreme effects on the comparison.

The dependence of the mean running time on various parameters is shown in the following figures (all indicated timings are wall-clock times):

- Fig. 4(a) shows the influence of the number of suppliers $|S|$. Since the Outer Linearization and the Scenario Enumeration Algorithm both iterate over all scenarios, their running times increase at least linearly with the number of scenarios. In contrast, the Iterative Cut Generation Algorithm uses an implicit representation of the scenario set. Therefore, it does not need to iterate over every scenario. However, the number of variables in the IPs increase.
- Fig. 4(b) illustrates the dependence on the number of demand nodes $|D|$. Since the number of scenarios does not depend on the number of demand nodes, the impact of this parameter on the running time is different from the impact of the number of suppliers. In particular the Outer Linearization Algorithm is less affected by an increase of the number of demand nodes. The variation of the running time of the Iterative Cut Generation Algorithm is mostly due to the randomness of the instances which influences the number of required iterations.

Table 1
NETGEN parameters used in generation, depending on the number of suppliers $|S|$ and the number of demand nodes $|D|$. All other parameters were set to 0.

Sources	Sinks	Nodes	Arcs	Total supply	Cap. skel. arcs	Max. arc cap.
$\|S\|$	$\|D\|$	$\|S\|+\|D\|$	$\frac{1}{2}\|S\|\cdot\|D\|$	10^5	100%	$\frac{3\cdot 10^5}{\|S\|\cdot\|D\|}$

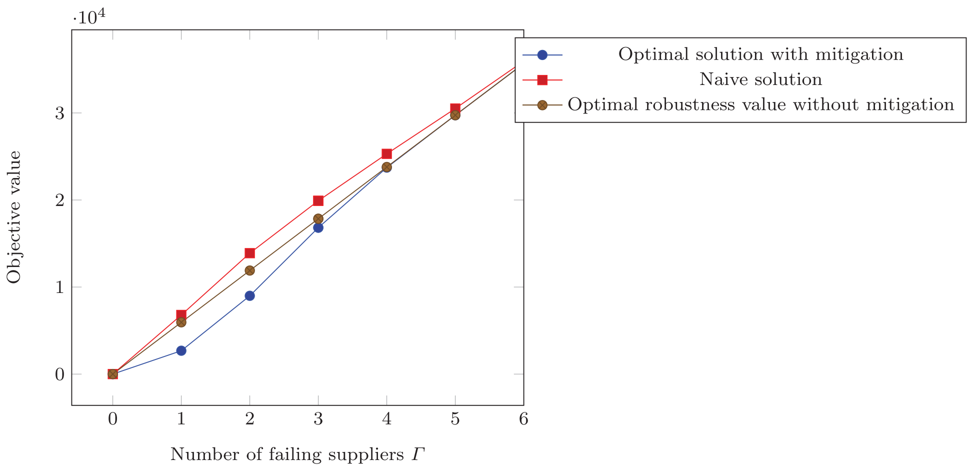


Fig. 2. Dependence of the objective value on the number of failing suppliers Γ. The chosen instance contains $|S| = 30$ suppliers. The total supply is 10^5.

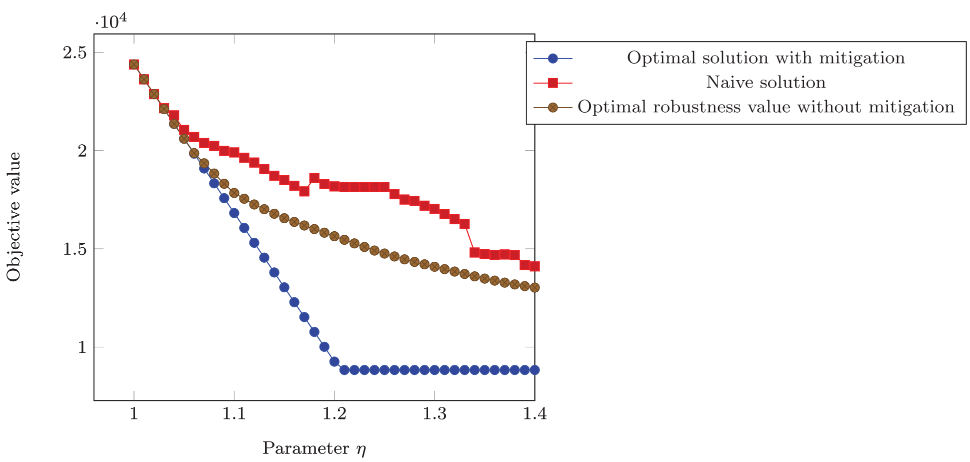


Fig. 3. Dependence of the objective value on the parameter η specifying the over-capacity of the suppliers. The chosen instance contains $|S| = 30$ suppliers, the number of failing suppliers was set to $\Gamma = 3$. The total supply is 10^5.

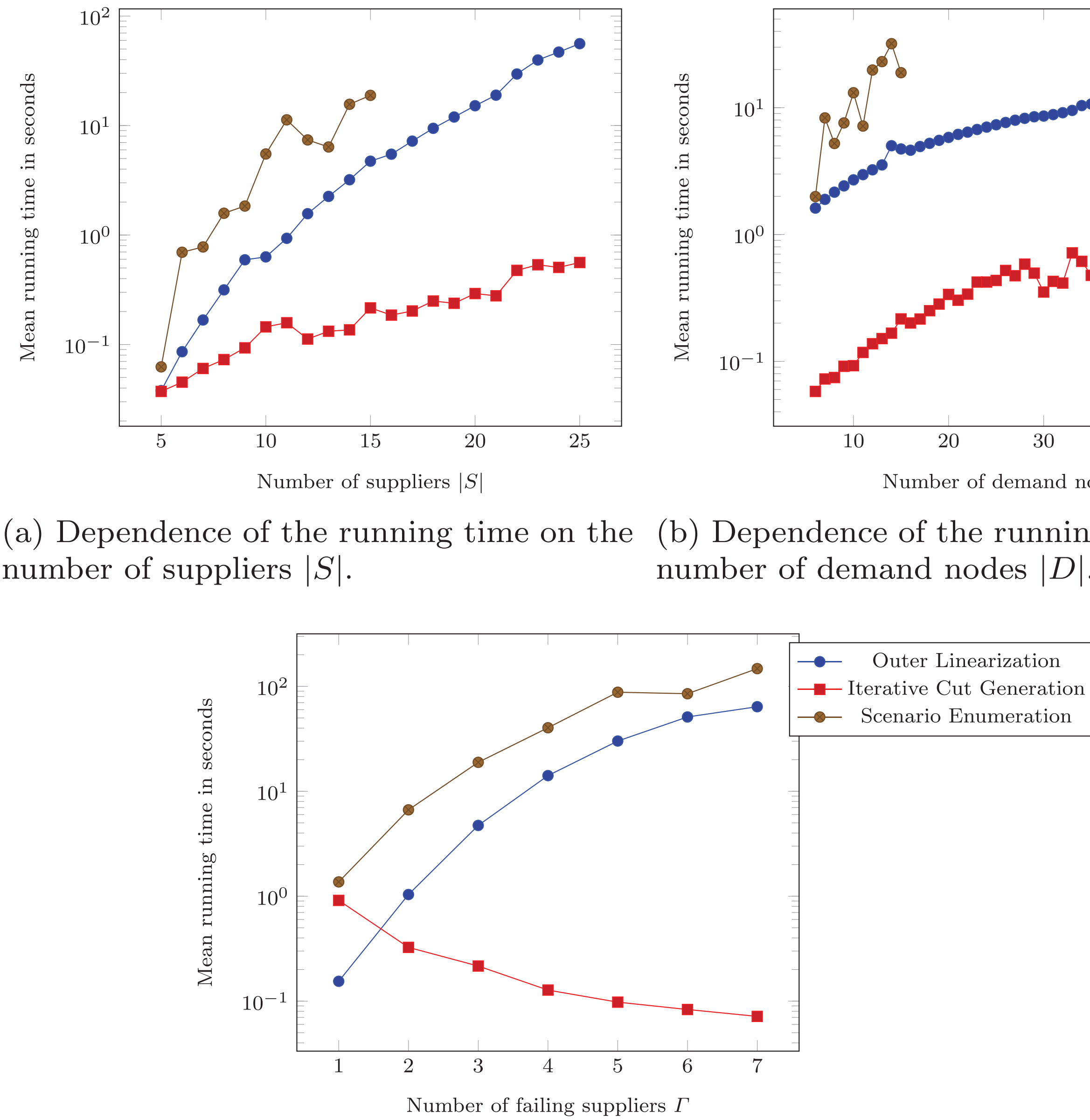
- In Fig. 4(c) the influence of the size of the scenario sets is illustrated. The number of considered scenarios grows exponentially with the parameter Γ up to Γ = 5, since the number of scenarios is given by $|X| = \binom{|S|}{\Gamma}$. As expected, the running time of the Outer Linearization and the Scenario Enumeration Algorithm increases since it is at least linear in the number of scenarios. However, the running time of the Iterative Cut Generation decreases with increasing Γ, since the found worst-case scenarios are more severe when they include more failing suppliers. Therefore the generated cuts are restricting the flows more strongly, decreasing the number of needed iterations.

By analyzing the dependence of the running time of the algorithms on various parameters, we get the following general results:

- The Iterative Cut Generation Algorithm is overall the fastest algorithm. However, for very small scenario sets, the Outer Linearization Algorithm is faster. For large scenario sets, the Iterative Cut Generation Algorithm is especially suited, since the running time does *not* increase when the number of suppliers considered in a failure scenario increases.
- The Scenario Enumeration Algorithm is generally the slowest one. However, the algorithm is more flexible than the others, since it allows to optimize with respect to an arbitrary consistent risk measure with a simple modification.
- The Outer Linearization Algorithm is the easiest to implement and analyze. It has however a large running time and an extremely large memory requirement.

As the Iterative Cut Generation Algorithm performs significantly better than the other algorithms, we can apply it to significantly larger instances. The following plots illustrate its running time behavior on larger instances. Fig. 5(a)–(c) shows the dependence of the running time on the number of suppliers $|S|$, number of demand nodes $|D|$ and number of failing suppliers Γ. The results indicate that the running time behavior becomes relatively stable, once the parameter crosses some threshold, i.e. for $|S| > 20$ suppliers, $|D| > 40$ demand nodes and $\Gamma \geq 4$ failing suppliers.

Fig. 6 illustrates the behavior of the Iterative Cut Generation Algorithm over time for a fixed instance. The plot shows that already after a fraction of the time (0.3 seconds), an almost optimal solution is found by the algorithm. The remaining time is mainly used to verify the optimality of this solution.



(a) Dependence of the running time on the number of suppliers $|S|$.

(b) Dependence of the running time on the number of demand nodes $|D|$.

(c) Dependence of the running time on the number of failing suppliers Γ.

Fig. 4. Comparison of the running time of the three algorithms, depending on multiple parameters of the instances. The default parameters used here are $|S| = 15$, $|D| = 15$, $\Gamma = 3$.

6. Extensions and outlook

The robust flow model can be extended in various ways. We present here briefly some of those extensions. In most cases, the Iterative Cut Generation Algorithm 2 can be adapted to also solve these extended problems by modifying the outer LP or the inner evaluation IP.

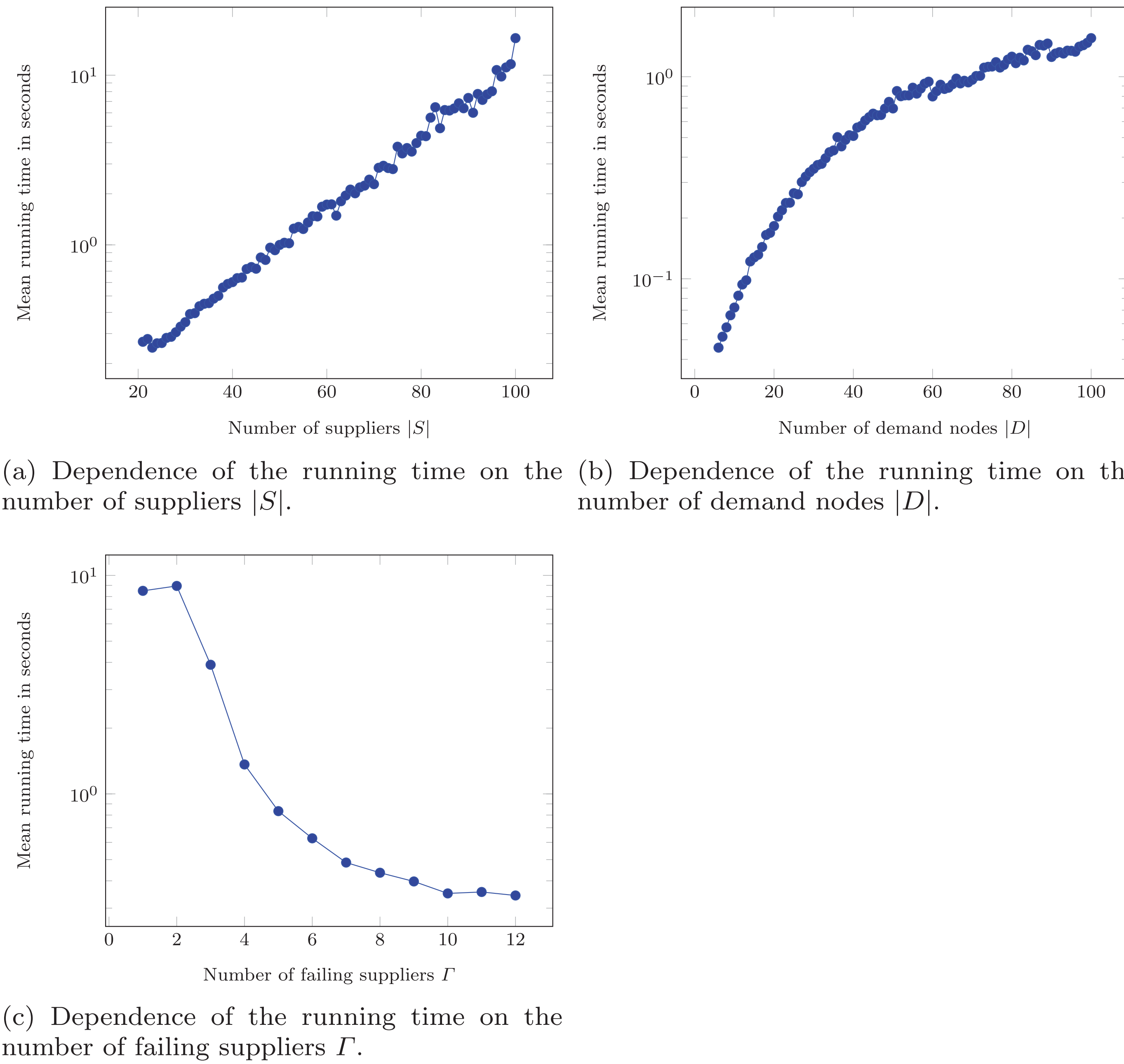
Different choices for the mitigation capacities: Instead of taking the residual capacities from the nominal flow, the mitigation capacities can be given as static values. In this way, we can model supply chains where depots with fixed inventories have been placed that can be used for mitigation.

Arc failures: We can also model the failure of arcs in the network instead of suppliers. In conjunction with this, also different network topologies can be used apart from a bipartite network, e.g. by including transshipment nodes.

Shipping delays: The flows in the network can be assumed to take some time to reach their target. This limits the amount of possible mitigation if the transfer time for the mitigation is longer than the time for the nominal flow.

Placement of depots: Additional capacities for mitigation can be installed by opening some depots out of a given set of possible locations. For these depot capacities, a budget is given. The goal is to find an optimal choice of the depots together with an optimal flow.

Multicriteria optimization: The objective function of robustness can be combined with other criteria, like costs. For example, an upper bound for the flow costs can be inserted as a hard constraint in the outer LP.



(a) Dependence of the running time on the number of suppliers $|S|$.

(b) Dependence of the running time on the number of demand nodes $|D|$.

(c) Dependence of the running time on the number of failing suppliers Γ.

Fig. 5. Dependence of the running time of the Iterative Cut Generation Algorithm on various parameters. The default parameters used here are: $|S| = 30$, $|D| = 30$, $\Gamma = 10$.

Our work shows that it is possible to take second stage mitigation already into account when optimizing supply chains. Our approach is able to handle large scenario sets by using an implicit description of the scenario set. The approach offers great flexibility, since many additional constraints can be modeled into the LP (1) without extensive modifications to the algorithm. The results also indicate that fully adjustable robust optimization can be solved in practice within a reasonable running time although evaluating the robustness is NP-hard.

Future work might consider the above extensions in more detail. The combination with a more detailed model of the sources of failure can lead to further insights. The structural properties of supply chains that are optimal with respect to the robustness measure might be interesting to supply chain practitioners. Also, a more detailed study of the factors that influence the running time of the Iterative Cut Generation Algorithm is warranted.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to two anonymous referees for suggestions that improved the clarity of the paper and for drawing attention to a small modification in the Iterative Cut Generation Algorithm that significantly improved its performance.

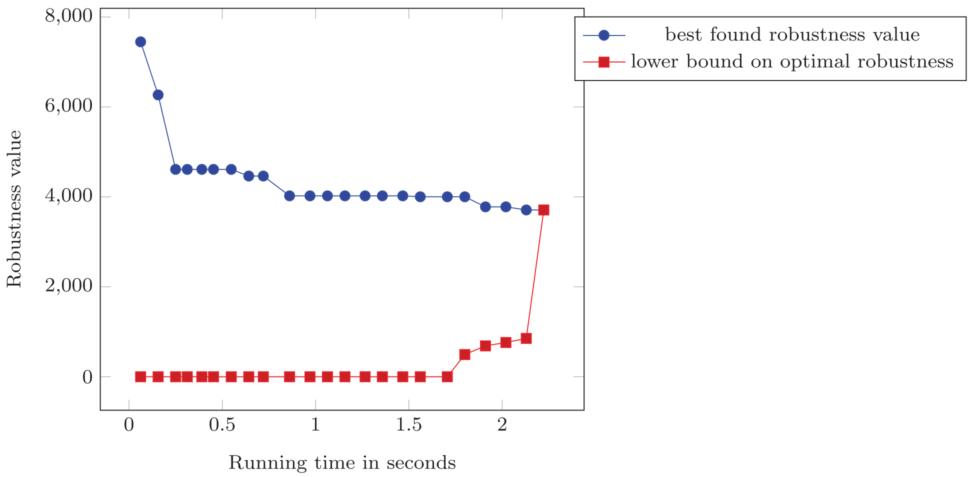


Fig. 6. The evolution of the robustness of the current solutions over the iterations of the Iterative Cut Generation Algorithm. Note that a robustness value closer to 0 is better. Here, the run of the Iterative Cut Generation Algorithm for a randomly generated, medium-sized instance with $|S| = 30$, $|D| = 30$ and $\Gamma = 4$ is shown. The upper bound corresponds to the objective value of the outer LP. Also the best found robustness value, i.e. the evaluated robustness value of the current best solution, is displayed.

Appendix A: non-optimality of affine decision rules

Proof of Proposition 1. We show that restricting the mitigation to be an affine function of the scenario yields a non-optimal mitigation. As an affine dependence, we require that the mitigation f' is affine in the vector $x = (x_s)_{s\in S}$ representing the scenario of failing suppliers, i.e. there should exist a matrix A and a vector d such that

$$f' = Ax + d$$

holds for every scenario $x \in X$.

Consider the network depicted in Fig. A1 with suppliers $S = \{s_1, \dots, s_4\}$ and demand nodes $D = \{d_1, \dots, d_4\}$. We use the scenario set $X = X(2)$ given as the Γ-robust scenario set with $\Gamma = 2$.

Consider now the failure scenario (s_2, s_3). Then, a shortage of 1 each is created at the demand nodes d_2 and d_3. Since there is no arc from s_4 to d_3, the remaining capacity of s_4 can only be used to mitigate the shortage at d_2. Hence, in an optimal mitigation f', we have $f'_{s_4,d_2} = 1$. Thus, there only remains shortage at d_3, which can be mitigated by d_1 by setting $f'_{s_1,d_3} = 1$. The other entries of the mitigation vector f' are 0.

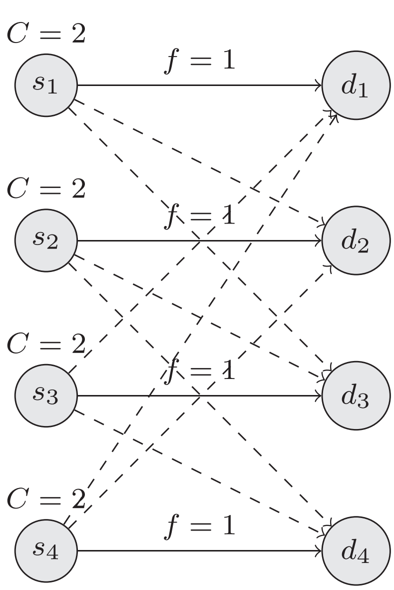


Fig. A1. A network together with a nominal flow f. The flow over arcs not labeled by f is 0. The capacity of each arc is unlimited, however each supplier has a capacity limit of 2. The arcs are chosen in such a way that s_i is connected to d_i, d_{i+1} and d_{i+2}, where indices are taken modulo 4.

Thus, f' mitigates all 2 units of shortage. Note that the arguments show that an optimal mitigation must have this form. Every other feasible mitigation has a unmitigated shortage greater than 0.

We consider now what the optimal mitigation is in some other shortage scenarios. We will use this information, to show that an affine decision rule is not optimal in this case.

- In the scenario (s_1, s_3) the unique optimal mitigation is given by

  $$f'_{s_2,d_3} = 1, \quad f'_{s_3,d_1} = 1,$$

  and all other entries of f' set to 0.
- In scenario (s_2, s_4) the nonzero entries of the unique optimal mitigation are given by:

  $$f'_{s_1,d_2} = 1, \quad f'_{s_3,d_4} = 1.$$
- In scenario (s_1, s_4) it is given by:

  $$f'_{s_2,d_4} = 1, \quad f'_{s_3,d_4} = 1.$$

Suppose that an affine decision rule yields an optimal solution for this example. Then, since the optimal mitigation is always unique for each considered scenario, the output of the affine decision rule must coincide with the mitigation vectors given above. Let the affine function h represent this affine decision rule, i.e.

$$h : \mathbb{R}^4 \to \mathbb{R}^{4\times 4}, x \mapsto f'$$

maps a vector $x = (x_s)_{s\in S}$ indicating the failure scenario to a vector $f' = (f'_{s,d})_{s\in S, d\in D}$ representing the mitigation flow.

By writing a scenario x as

$$x = \begin{pmatrix} x_{s_1} \\ x_{s_2} \\ x_{s_3} \\ x_{s_4} \end{pmatrix},$$

we have the following equality since h is affine:

$$h\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = h\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} + h\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} - h\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{A.1}$$

Suppose now that the function h would always yield the optimal mitigation in each scenario. Then, the following must hold by our analysis of the optimal mitigation in the scenarios considered above, using the notation e_{s_i,d_j} for the unit vector of the component (s_i, d_j):

$$h\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = e_{s_1,d_3} + e_{s_4,d_2}, \qquad h\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = e_{s_2,d_3} + e_{s_3,d_1},$$

$$h\begin{pmatrix}0\\1\\0\\1\end{pmatrix} = e_{s_1,d_2} + e_{s_3,d_4}, \quad h\begin{pmatrix}1\\0\\0\\1\end{pmatrix} = e_{s_2,d_4} + e_{s_3,d_1}.$$

By plugging this in, we see directly that this contradicts the affinity condition (A.1). Hence, the assumption that h always gives an optimal mitigation lead to a contradiction. Therefore, affine decision rules are *not* optimal in this case.

We now analyze the quality that an optimal affine decision rule provides. Consider the failure of a supplier s_i. Then, a shortage is created at demand node d_i. The mitigation at demand node d_i can be done from the suppliers s_{i-2} and s_{i-1}, where indices are taken modulo 4. Thus, the optimal affine function h should fulfill

$$h(e_{s_i}) = \alpha \cdot e_{s_{i-2},d_i} + \alpha \cdot e_{s_{i-1},d_i},$$

for some $\alpha \in \mathbb{R}_{\geq 0}$. The choice of the same factor α in front of e_{s_{i-2},d_i} and e_{s_{i-1},d_i} follows from the symmetry of the network: Locally, with respect to single suppliers, the nodes s_{i-2} and s_{i-1} are equivalent.

Similarly by symmetry, we can assume that the value of α is the same for every supplier s_i. We now analyze the optimal value for α. Obviously, α should be chosen as large as possible, since a larger value of α increases the mitigation amount. We have that for the scenario of a failure of suppliers s_1 and s_2, the corresponding mitigation given by the affine function,

$$h(e_{s_1} + e_{s_2}) = h(e_{s_1}) + h(e_{s_2}) = \alpha e_{s_3,d_1} + \alpha e_{s_4,d_1} + \alpha e_{s_4,d_2} + \alpha e_{s_1,d_2},$$

should be feasible. By the capacity requirement of supplier s_4, the following must hold for feasibility:

$$f_{s_4,d_4} + 2\alpha \leq C_{s_4},$$

i.e.

$$1 + 2\alpha \leq 2 \iff \alpha \leq \frac{1}{4}.$$

Thus, the optimal choice is $\alpha = \frac{1}{4}$. The corresponding mitigated amount in each scenario is then $4\alpha = 4 \cdot \frac{1}{4} = 1$. Thus, since the shortage was 2, the unmitigated shortage is 1 for the optimal affine decision rule.

In contrast, the unmitigated shortage for the optimal mitigation is just 0. □

Example 2 (Special case of optimality of affine decision rules)**.** In the case that the scenario set consists of linearly independent scenarios, affine decision rules are optimal, since we can choose an affine function in the scenario x such that a desired value is met for every scenario $x \in X$ contained in the scenario set. In particular, this is the case if the scenario set consists only of scenarios with one failing supplier, Formally, we can find a matrix A and vector d such that for every $x \in X$ we have that

$$Ax + d = f^*(x)$$

where $f^*(x)$ is the optimal mitigation in scenario x. This is possible, since the family of vectors $(x)_{x \in X}$ is linearly independent by the given assumption.

Hence, if the family of scenarios $(x)_{x \in X}$ is linearly independent, the affine decision rule is always optimal. Note that in this case however, the size of the scenario set must be bounded by $|X| \leq |S|$ in the number of suppliers. Thus, an approach which explicitly constructs an optimal mitigation for each scenario also leads to an efficiently computed optimal solution.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 2014. Network Flows: Theory, Algorithms and Applications. Pearson.
Alderson, D.L., Brown, G.G., Carlyle, W.M., Wood, R.K., 2011. Solving defender-attacker-defender models for infrastructure defense. In: Proceedings of the Twelfth INFORMS Computing Society Conference. INFORMS, pp. 28–49. doi:10.1287/ics.2011.0047.
Alderson, D.L., Brown, G.G., Carlyle, W.M., Wood, R.K., 2018. Assessing and improving the operational resilience of a large highway infrastructure system to worst-case losses. Transp. Sci. 52 (4), 1012–1034. doi:10.1287/trsc.2017.0749.
Alvarez-Miranda, E., Ljubić, I., Raghavan, S., Toth, P., 2014. The recoverable robust two-level network design problem. INFORMS J. Comput. 27 (1), 1–19. doi:10.1287/ijoc.2014.0606.
Aneja, Y.P., Chandrasekaran, R., Nair, K.P.K., 2001. Maximizing residual flow under an arc destruction. Networks 38 (4), 194–198. doi:10.1002/net.10001.
Atamtürk, A., Zhang, M., 2007. Two-stage robust network flow and design under demand uncertainty. Oper. Res. 55 (4), 662–673. doi:10.1287/opre.1070.0428.
Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust Optimization, 28. Princeton University Press.
Ben-Tal, A., Golany, B., Nemirovski, A., Vial, J.-P., 2005. Retailer-supplier flexible commitments contracts: a robust optimization approach. Manuf. Serv. Oper. Manag. 7 (3), 248–271. doi:10.1287/msom.1050.0081.
Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4 (1), 238–252. doi:10.1007/BF01386316.
Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2004. Adjustable robust solutions of uncertain linear programs. Math. Program. 99 (2), 351–376. doi:10.1007/s10107-003-0454-y.
Bertsimas, D., Litvinov, E., Sun, X.A., Zhao, J., Zheng, T., 2013. Adaptive robust optimization for the security constrained unit commitment problem. IEEE Trans. Power Syst. 28 (1), 52–63. doi:10.1109/TPWRS.2012.2205021.
Bertsimas, D., Nasrabadi, E., Stiller, S., 2013. Robust and adaptive network flows. Oper. Res. 61 (5), 1218–1242. doi:10.1287/opre.2013.1200.
Birge, J., Louveaux, F., 2011. Introduction to Stochastic Programming. Springer.
Brown, G., Carlyle, M., Salmerón, J., Wood, K., 2006. Defending critical infrastructure. Interfaces 36 (6), 530–544. doi:10.1287/inte.1060.0252.
Brown, G.G., Carlyle, W.M., Wood, R.K., 2008. Department of Homeland Security Bioterrorism Risk Assessment. National Academies Press, Washington, D.C. doi:10.17226/12206.
Cacchiani, V., Jünger, M., Liers, F., Lodi, A., Schmidt, D.R., 2016. Single-commodity robust network design with finite and hose demand sets. Math. Program. 157 (1), 297–342. doi:10.1007/s10107-016-0991-9.
Cappanera, P., Scaparra, M.P., 2011. Optimal allocation of protective resources in shortest-path networks. Transp. Sci. 45 (1), 64–80. doi:10.1287/trsc.1100.0340.
Diessel, E., 2019. Risk aware flow optimization and application to logistics networks, Master Thesis Technische Universität Kaiserslautern, Department of Mathematics.
Downey, R.G., Fellows, M.R., 1999. Parameterized Complexity. Monographs in Computer Science. Springer.
Du, D., Chandrasekaran, R., 2007. The maximum residual flow problem: NP-hardness with two-arc destruction. Networks 50 (3), 181–182. doi:10.1002/net.20188.
Garg, M., Smith, J.C., 2008. Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios. Omega 36 (6), 1057–1071. doi:10.1016/j.omega.2006.05.006.
Gurobi Optimization, LLC, 2019. Gurobi Optimizer Reference Manual.
Hagberg, A.A., Schult, D.A., Swart, P.J., 2008. Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (Eds.), Proceedings of the Seventh Python in Science Conference, pp. 11–15.
Hamdi, F., Ghorbel, A., Masmoudi, F., Dupont, L., 2018. Optimization of a supply portfolio in the context of supply chain risk management: literature review. J. Intell. Manuf. 29 (4), 763–788. doi:10.1007/s10845-015-1128-3.
Ho, W., Zheng, T., Yildiz, H., Talluri, S., 2015. Supply chain risk management: a literature review. Int. J. Prod. Res. 53 (16), 5031–5069. doi:10.1080/00207543.2015.1030467.
Klingman, D., Napier, A., Stutz, J., 1974. Netgen: a program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. Manag. Sci. 20 (5), 814–821. doi:10.1287/mnsc.20.5.814.
Lozano, L., Smith, J.C., 2017. A backward sampling framework for interdiction problems with fortification. INFORMS J. Comput. 29 (1), 123–139. doi:10.1287/ijoc.2016.0721.
Marandi, A., den Hertog, D., 2018. When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent? Math. Program. 170 (2), 555–568. doi:10.1007/s10107-017-1166-z.
McCormick, G.P., 1976. Computability of global solutions to factorable nonconvex programs: part I — convex underestimating problems. Math. Program. 10 (1), 147–175. doi:10.1007/BF01580665.
Phillips, C.A., 1993. The network inhibition problem. In: Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing, pp. 776–785.
Prince, M., Smith, J.C., Geunes, J., 2013. A three-stage procurement optimization problem under uncertainty. Nav. Res. Logist. 60 (5), 395–412. doi:10.1002/nav.21541.
Roberto, M.A., Levesque, L.C., 2004. Managing risk to avoid supply-chain breakdown. MIT Sloan Manag. Rev. 46 (1), 53–60.
Rockafellar, R.T., Uryasev, S., 2000. Optimization of conditional value-at-risk. J. Risk 2, 21–41. doi:10.21314/JOR.2000.038.
Sadeghi, S., Seifi, A., Azizi, E., 2017. Trilevel shortest path network interdiction with partial fortification. Comput. Ind. Eng. 106, 400–411. doi:10.1016/j.cie.2017.02.006.
Schlenker, N., 1989. Capacitated Network Generator Netgen. URL: http://elib.zib.de/pub/mp-testdata/generators/netgen/index.html.
Simchi-Levi, D., Wang, H., Wei, Y., 2018. Increasing supply chain robustness through process flexibility and inventory. Prod. Oper. Manag. 27 (8), 1476–1491. doi:10.1111/poms.12887.
Simchi-Levi, D., Wang, H., Wei, Y., 2019. Constraint generation for two-stage robust network flow problems. INFORMS J. Optim. 1 (1), 49–70. doi:10.1287/ijoo.2018.0003.
Smith, J.C., Lim, C., Sudargho, F., 2007. Survivable network design under optimal and heuristic interdiction scenarios. J. Glob. Optim. 38 (2), 181–199. doi:10.1007/s10898-006-9067-3.

Thiele, A., Terry, T., Epelman, M., 2010. Robust linear optimization with recourse. IOE Technical Report TR09-01. University of Michigan.
Wood, R.K., 1993. Deterministic network interdiction. Math. Comput. Model. 17 (2), 1–18. doi:10.1016/0895-7177(93)90236-R.
Yanikoğlu, I., Gorissen, B.L., den Hertog, D., 2018. A survey of adjustable robust optimization. Eur. J. Oper. Res. 277 (3), 799–813. doi:10.1016/j.ejor.2018.08.031.
Yuan, W., Zhao, L., Zeng, B., 2014. Optimal power grid protection through a defender–attacker–defender model. Reliab. Eng. Syst. Saf. 121, 83–89. doi:10.1016/j.ress.2013.08.003.
Zeng, B., Zhao, L., 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. Oper. Res. Lett. 41 (5), 457–461. doi:10.1016/j.orl.2013.05.003.
Zhen, J., den Hertog, D., Sim, M., 2018. Adjustable robust optimization via Fourier–Motzkin elimination. Oper. Res. 66 (4), 1086–1100. doi:10.1287/opre.2017.1714.