[EURO Journal on Computational Optimization 9 (2021) 100002](https://doi.org/10.1016/j.ejco.2020.100002)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

EURO Journal on Computational Optimization

journal homepage: [www.elsevier.com/locate/ejco](http://www.elsevier.com/locate/ejco)

Robust flows with adaptive mitigation

Heiner Ackermann[a](#_bookmark0), Erik Diessel[a](#_bookmark0),[∗](#_bookmark2), Sven O. Krumke [b](#_bookmark1)

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*

a r t i c l e i n f o a b s t r a c t

*MSC:*

90B06

90C11

90C47

90C35

*Keywords:*

Adjustable robust optimization Supply chain

Cut-generation Network flows

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.

# Introduction

In supply chains, shortages caused by supply failures can have a sig- nificant impact, leading to large amount of lost sales and a possible rep- utation loss due to unfulfilled demand ([Roberto and Levesque, 2004](#_bookmark61)). Therefore it is important to hedge against these risks when making de- cisions affecting the supply chain configuration ([Ho et al., 2015](#_bookmark55)). Tradi- tional optimization models consider only the associated risk of shortages ([Hamdi et al., 2018](#_bookmark54)) 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 con- straints. Ignoring this possibility leads to an overly pessimistic view of the risk and in suboptimal decisions with respect to realistic risk mea- sures.

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](#_bookmark43)), 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 arbi- trarily 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](mailto:heiner.ackermann@itwm.fraunhofer.de) (H. Ackermann), [erik.diessel@itwm.fraunhofer.de](mailto:erik.diessel@itwm.fraunhofer.de) (E. Diessel), [krumke@mathematik.uni-kl.de](mailto: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

2192-4406/© 2020 The Author(s). Published by Elsevier Ltd on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

* 1. *Previous work*

Robust Optimization deals with optimization problems where some of the coeﬃcients, in the inequalities describing the feasible set, are un- certain. In contrast to stochastic programming, no probability distribu- tion 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{*𝑐𝑇 𝑥* ∣ *𝐴*(*𝜁* )*𝑥* ≤ *𝑑* for all *𝜁* ∈ Ę}*,*

*𝑥*

and *A*(*𝜁* ) is a matrix depending on the realized uncertainty *𝜁* . We refer where Ę represents the set of possible realizations of the uncertainty

to the textbook by [Ben-Tal et al. (2009)](#_bookmark36) 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)](#_bookmark39). 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 pos- sible realizations of the uncertain parameters. A recent survey was given by [Yanikoğlu et al. (2018)](#_bookmark74). The adjustable robust counterpart has the form

min{*𝑐𝑇 𝑥* ∣ *𝐴*(*𝜁* )*𝑥* + *𝐵𝑦*(*𝜁* ) ≤ *𝑑* for all *𝜁* ∈ Ę}*,*

*𝑥,𝑦*(⋅)

where *x* is the first-stage decision and *y*(*𝜁* ) is the second-stage decision that is a function of the realized uncertainty *𝜁* ∈ Ę.

stages decisions *y* that depend aﬃnely on *𝜁* (Aﬃnely Adjustable Ro- In the literature, most algorithms deal with the case of second-

bust Optimization), i.e. there is a matrix *Q* and a vector *y*0 such that

*𝑦*(*𝜁* ) = *𝑦*0 + *𝑄𝜁* . This leads to easy decision rules for *y* (and often poly-

nomially 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](#_bookmark14). Results for the equivalence of adjustable and robust optimization in certain cases ([Marandi and den Hertog, 2018](#_bookmark58)), 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)](#_bookmark42) 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)](#_bookmark71) 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)](#_bookmark76) extend the algo- rithm to also generate new columns and apply it to an adjustable robust location-transportation problem. [Bertsimas et al. (2013a)](#_bookmark41) applied this method to solve an adjustable robust problem in unit commitment of electric power systems. [Simchi-Levi et al. (2019)](#_bookmark67) describe an algorithm to solve Robust Network Flow problems based on cuts generated with an IP. Recently, [Zhen et al. (2018)](#_bookmark73) proposed to solve Adjustable Robust Op- timization 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)](#_bookmark34) discuss a re-

[coverable version of the Two-Level Network Design problem. Garg and Smith (2008) consider a robust network design problem to ensure feasi-](#_bookmark50) ble multicommodity flows exists. [Simchi-Levi et al. (2018)](#_bookmark65) formulate a model for robust inventory optimization incorporating decisions on pro- duction quantities in the second stage. [Cacchiani et al. (2016)](#_bookmark45) as well as [Atamtürk and Zhang (2007)](#_bookmark37) and [Ben-Tal et al. (2005)](#_bookmark38) 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](#_bookmark60)). The problem of finding the worst-case scenario for a given flow in our robust setting can be inter- preted as a network interdiction problem: Find an interdiction strategy (the scenario) such that the unmitigated shortage is maximized. The par- ticular 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 Prob- lem considered by [Aneja et al. (2001)](#_bookmark35). In this problem, the goal is to find a flow such that the residual flow after an arc destruction is maxi- mized. 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 fail- ures ([Du and Chandrasekaran, 2007](#_bookmark51)).

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)](#_bookmark44). [Smith et al. (2007)](#_bookmark68) discuss a three-stage prob- lem 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 prob- lem, employing similar techniques as in our Iterative Cut Genera- tion [Algorithm 2](#_bookmark21). [Alderson et al. (2011)](#_bookmark69) discuss a tri-level network model in the context of infrastructure defense. They use a decompo- sition algorithm based on mixed integer programs with quadratic con- straints. [Alderson et al. (2018)](#_bookmark72) extend this model and analyze the re- sults of a detailed case-study. The study of [Brown et al. (2008)](#_bookmark46) ap- plies attacker-defender-attacker models to counter-terrorism efforts. [Yuan et al. (2014)](#_bookmark75) describe a defender-attacker-defender model for elec- trical power grids which they solve with a column-generation approach. [Lozano and Smith (2017)](#_bookmark56) discuss a backward sampling algorithm for solving tri-level network optimization problems with fortification. In the paper by [Prince et al. (2013)](#_bookmark62) a model for a three-stage procure- ment problem in supply chains is introduced, where an adversary can re- [move some of the capacities if no fortification is applied. Cappanera and Scaparra (2011) describe a tri-level model for optimizing the robustness](#_bookmark47) of a network with respect to shortest-path distances and heuristics for an eﬃcient solution.

* 1. *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 algo-

rithm that yields an exact optimal solution with respect to this 𝖭𝖯-hard objective function. For this problem, we compare three different algo- rithms, from a simple to a more advanced approach. The simplest is the Outer Linearization Algorithm that uses an explicit LP-reformulation of

′

∑ *𝑓*

*𝑠,𝑑*

*𝑑*∈*𝐷*

≤ *𝐶𝑠* − *𝑓*

*𝑑*∈*𝐷*

( ∑

*𝑠,𝑑*

)(1 − *𝑥𝑠* ) for all *𝑠* ∈ *𝑆* (2c)

the robustness. The Scenario Enumeration Algorithm maintains an ap- proximation of the robustness value as an LP. In each iteration, this

’

*𝑠,𝑑*

∑ *𝑓*

*𝑠*∈*𝑆*

≤ *𝑥𝑠* ⋅ *𝑓𝑠,𝑑* for all *𝑑* ∈ *𝐷* (2d)

*𝑠*∈*𝑆*

∑

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,

′

*𝑠,𝑑*

*𝑓*

≥ 0 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷* (2e)

The constraints [(2b)](#_bookmark12) ensure that the mitigation flow over the arc

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 Algo- rithm allows to solve medium-sized instances eﬃciently. We also de- scribe how the algorithm can be adapted for solving various extensions of the model.

This paper is organized as follows: In [Section 2](#_bookmark5) we formally define the basic model. In [Section 3](#_bookmark15) we show that evaluating a given flow for its robustness in the adaptive setting is NP-hard. [Section 4](#_bookmark20) presents our al- gorithms, the corresponding experimental results are given in [Section 5](#_bookmark24). We conclude in [Section 6](#_bookmark29) where we also discuss extensions of the basic model.

(*s,d*) is 0 whenever the supplier *s* has a failure, i.e. *𝑥𝑠* = 1. Otherwise,

the mitigation is bounded by the residual capacity *𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* . Similarly,

[(2c)](#_bookmark3) ensures that the mitigation flow does not exceed the residual capac-

ity of the supplier. Failures do not need to be considered here because of [(2b)](#_bookmark12). Constraint [(2d)](#_bookmark4) 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 corre- sponds to a capacitated transportation problem which can be solved in polynomial time ([Ahuja et al., 2014](#_bookmark70)).

**Remark 1.** The formulation of the linear program in [Definition 1](#_bookmark10) has

constraint [(2b)](#_bookmark12) already implies *𝑓* ′ = 0 for every *s* ∈ *S* with *𝑥𝑠* = 1*,* the a large influence on the performance of the later algorithms. Since the

*𝑠,𝑑*

constraint [(2c)](#_bookmark3) could be reformulated as

∑ *𝑓*

∑

# Our model

′

*𝑠,𝑑*

*𝑑*∈*𝐷*

≤ *𝐶𝑠* − *𝑓𝑠,𝑑*

*𝑑*∈*𝐷*

We consider a complete bipartite network *𝐺* = (*𝑉* = *𝑆* ∪ *𝐷, 𝐸*) of sup- pliers *S* and demand nodes *D*, where a single commodity of material is

distributed. Each supplier *s* ∈ *S* has a maximal production capacity *Cs* ≥ 0 and the arc (*s, d*) ∈ *E* between *s* and *d* has capacity *cs,d* ≥ 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 bipar- tite graph, since a non-existing arc can be represented by one with zero

capacity. Each demand node *d* ∈ *D* has an associated demand *bd* ≥ 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:

∑ *𝑓𝑠,𝑑* = *𝑏𝑑* for all *𝑑* ∈ *𝐷* (1a)

*𝑠*∈*𝑆*

∑ *𝑓𝑠,𝑑* ≤ *𝐶𝑠* for all *𝑠* ∈ *𝑆* (1b)

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* = (*𝑥𝑠* )*𝑠*∈*𝑆* ∈ *𝑋* gives the status of every supplier. In this way we can

consider scenarios with simultaneous failures of multiple suppliers. The

the difference of the mitigated amount *𝑧*∗ (*𝑥, 𝑓* ) and the total shortage unmitigated shortage in scenario *x* with the mitigation flow *f*′ is given by Σ*s* ∈ *S*Σ*d* ∈ *Dxsfs,d* . We now have all prerequisites for defining the main

*𝑁*

subject of our paper.

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

( ∑ ∑ )

*𝑑*∈*𝐷*

0 ≤ *𝑓𝑠,𝑑* ≤ *𝑐𝑠,𝑑* for all (*𝑠, 𝑑*) ∈ *𝐸* (1c)

max*𝑓* min*𝑥*∈*𝑋 𝑧*∗ (*𝑥, 𝑓* ) − *𝑠*∈*𝑆 𝑑*∈*𝐷 𝑥𝑠 𝑓𝑠,𝑑*

s.t. nominal flow constraints (1) for *𝑓*

*𝑁*

(3)

Subsequently we refer to the conditions in [(1)](#_bookmark7) as the nominal con- straints. The failure scenario is given as a vector *x* with a component

*xs* ∈ {0, 1} for each supplier *s* ∈ *S*. The entry *xs* is 1 when supplier *s*

between supplier *s* and demand node *d* ∈ *D*. The mitigation flow *𝑓* ′ has a failure and 0 otherwise. This means that a flow of *xsfs,d* is lost

*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 re- striction is due to the fact that already agreed on shipments often cannot be changed anymore. Procurement contracts are often binding for rela- tively long time frames and produced material is in many cases specifi- cally 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 nomi-

We call the objective value in [(3)](#_bookmark8) 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.

resentation of a scenario set is given by a weight matrix *𝐴* ∈ ℝ*𝑘*× *𝑆* and **Definition 3** (Implicit representation of a scenario set)**.** An implicit rep- a parameter vector Γ ∈ ℝ*𝑘,* where *𝑘* ∈ ℕ. The corresponding scenario set

| |

is then given by:

*𝑋*(*𝐴,* Γ) ∶= {*𝑥* ∈ {0*,* 1}|*𝑆*| *𝐴𝑥* ≤ Γ}*.*

|

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

nal 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:

*𝑋*(Γ) ∶=

{*𝑥* ∈ {0*,* 1}

*𝑆*

| *𝑠*∈*𝑆*

| | | ∑

*𝑥𝑠* ≤ Γ}*.*

**Definition 1** (Mitigation problem)**.**

*𝑧*∗ (*𝑥, 𝑓* ) ∶= max ∑ ∑ *𝑓* ′

(2a)

**Example 1** (Illustrative solution of an instance)**.** In [Fig. 1](#_bookmark13), an example instance is shown together with an optimal mitigation.

*Non-optimality of affine decision rules*. A decision rule in the context

*𝑁*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑠,𝑑*

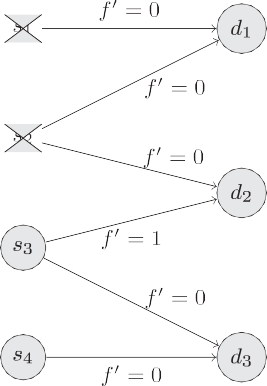
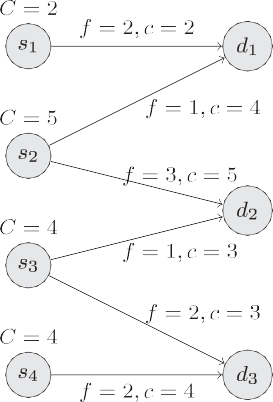
of adjustable robust optimization is a mapping from the scenario vari-

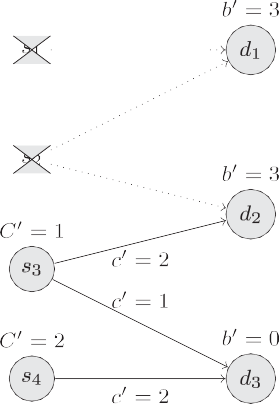
s.t. *𝑓* ′

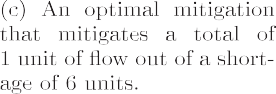
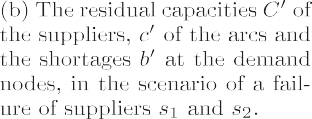
*𝑠,𝑑*

≤ (*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* )(1 − *𝑥𝑠* ) for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷* (2b)

ables to second-stage decisions. In our case, this would correspond to a

**Fig. 1.** Example instance of the Robust Flow prob- lem.





mapping from the scenario set *X* to mitigation flows *f*′. In the literature, Adjustable Robust Optimization problems are often tackled by restrict-

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

ing the decision rules to aﬃne functions in the scenario variables to achieve an approximation ([Ben-Tal et al., 2004](#_bookmark39)).

The following proposition shows that the restriction to aﬃne de-

(*𝑐*

*𝑠*∈*𝑆*

*𝑠,𝑑*

– *𝑓*

*𝑠,𝑑*

)(1 − *𝑥𝑠* )*.*

cision 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 short- age with fully adjustable mitigation is 0 whereas the restriction to an affine*

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

*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.*

(*𝑠*∑∈*𝑆*

*𝑥𝑠* ⋅ *𝑓𝑠,𝑑* −

*𝑠*∑∈*𝑆*

(*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* )(1 − *𝑥𝑠* ))+

A proof of [Proposition 1](#_bookmark14) can be found in the appendix.

# Hardness

( *𝑠*∑∈*𝑆*

*𝑓𝑠,𝑑* −

*𝑠*∑∈*𝑆*

*𝑐𝑠,𝑑* +

*𝑠*∑∈*𝑆*

*𝑐𝑠,𝑑 𝑥𝑠*

)+ *,*

*⏟⏟⏟*

=

=*𝑏𝑑* =const.

*⏟⏟⏟*

=const.

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.

Evaluation Problem consists of a network and a parameter Γ. For the **Definition 4** (Evaluation Problem)**.** An instance of the Flow Robustness

given network and nominal flow *f*, the goal is to compute

min (*𝑧*∗ (*𝑥, 𝑓* ) − ∑ ∑ *𝑥 𝑓* )*.*

*𝑥*∈*𝑋*(Γ)

*𝑁*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑠 𝑠,𝑑*

where the expression (*𝑔*)+ ∶= max{0*, 𝑔*} denotes the nonnegative part

of *g*.

Since the sums *𝑠*∈*𝑆 𝑓𝑠,𝑑* = *𝑏𝑑* and Σ*s* ∈ *Scs,d* are constant and inde-

∑

pendent of *f*, the unmitigated amount in this case is *independent* of *f*. If

we assume that the flow is feasible, we also have *fs,d* ≤ *cs,d* . Thus the expression above can be rewritten as

((∑ *𝑐𝑠,𝑑 𝑥𝑠* ) − *𝑎𝑑* )+ *,*

∑

*𝑠*∈*𝑆*

For this problem we will show 𝖭𝖯-hardness.

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

where *𝑎𝑑* ∶= *𝑠*∈*𝑆 𝑐𝑠,𝑑* − *𝑏𝑑* ≥ 0 is the constant corresponding to the de- mand *bd* and capacities *cs,d* for the considered demand node *d* ∈ *D*. Note that each possible value of 0 ≤ *ad* ≤ Σ*s* ∈ *Scs,d* can be obtained by choosing

nodes *d* ∈ *D*, we obtain the total unmitigated shortage, i.e. the negative appropriate values for the demand *bd* . By summing up over all demand

of the objective function in [Definition 4](#_bookmark16)

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 maxi-

∑ ((∑

*𝑐𝑠,𝑑 𝑥𝑠* )

+

– *𝑎𝑑*

)

= −(*𝑧*∗ (*𝑥, 𝑓* ) − ∑ ∑

*𝑥𝑠 𝑓𝑠,𝑑* )*.*

*𝑑*∈*𝐷 𝑠*∈*𝑆*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

mization problem. Using the representation, this can in turn be reduced to the Evaluation Problem.

*𝑁*

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

∑ *𝑥𝑠* ⋅ *𝑓𝑠,𝑑 .*

*𝑠*∈*𝑆*

Hence, because maximization of the negative objective is equiva- lent to minimization of the original objective, the question of finding a worst-case scenario according to [Definition 4](#_bookmark16) is equivalent to solving the maximization problem

max ∑ ((∑ *𝑐𝑠,𝑑 𝑥𝑠* ) − *𝑎𝑑* )+ *.* (4)

*𝑥*∈*𝑋*(Γ) *𝑑*∈*𝐷*

*𝑠*∈*𝑆*

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](#_bookmark17).

being positive is that 3Γ ⋅ *𝑥𝑣 >* 3Γ − 2*,* i.e. *𝑥𝑣 >* 3Γ−2 . We set *𝐶* ∶=

{*𝑣* ∈ *𝑉* ∣ *𝑥𝑣 >* 3Γ−2 } and show that *C* is a clique of size Γ. Since

3Γ

3Γ

Γ ≥ ∑ *𝑥𝑣 >* |*𝐶*| ⋅ 3Γ − 2 *,*

**Algorithm 1:** Scenario Enumeration

*𝑣*∈*𝑉*

3Γ

Initialize (∗) to be the following LP:

max *𝑡*

s.t. nominal flow constraints (1) for *𝑓*

we get from the assumption Σ*v* ∈ *Vxv* ≤ Γ the bound

Γ 3Γ = Γ+ 2 + 4 *<* Γ+ 1*,*

*𝑡* ≤ 0

where we assumed w.l.o.g. that Γ ≥ 3. Since

|*𝐶*| *<*

⋅ 3Γ − 2

3 9Γ − 6

Solve (∗), let *𝑓* ∗ be the corresponding optimal solution and let

**repeat**

∗

|*C*| ≤ Γ. Every individual term is bounded in the| fo|llowing way:

*𝑡* be its value

*𝐶 ,* Γ ∈ ℕ*,* this implies

*𝑣*

*𝑢*

3Γ ⋅ *𝑥* + ∑ *𝑥* − (4Γ − 2) ≤ 3Γ + Γ− 1 − (4Γ − 2) = 1*.*

mitigation in scenario *𝑥* with the current *𝑓* ∗ according to Definition 5; let *𝜈*∗(*𝑥*) be its value

**foreach** *scenario 𝑥* **do**

Find a maximal *𝑠*0 − *𝑑*0 flow in the network for the

Let (*𝑈, 𝑊* ) be the corresponding minimum cut

{

Because of our previous considerations there are at most |*C*| ≤ Γ 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* ∈ *C*)

*𝑢*∶(*𝑣,𝑢*)∈*𝐸*

set for *𝑑* ∈ *𝐷*, *𝑠* ∈ *𝑆*:*𝑦𝑑*

{0*, 𝑠* ∈ *𝑈,*

*𝜉*

∶=

*𝛽*

∶= 1*, 𝑑* ∈ *𝑈,*

0*, 𝑑* ∈ *𝑊 ,*

∶= max{1 − *𝜉* − *𝑦 ,* 0}

= Γ.

two nodes *v, w* ∈ *C* such that (*v, w*)∉*E*. By the definition of *C*, *𝑥𝑤 >* We now show that *C* is a clique. Assume the contrary, i.e. there are

and |*𝐶*|

3Γ−2 *>* 0. Thus, the term corresponding to *xv* has the following upper

*𝑠* 1*, 𝑠* ∈ *𝑊*

*𝑠,𝑑*

*𝑠 𝑑*

3Γ

bound

∑

Add the constraint

*𝑡* ≤ ∑ ∑(*𝑐* − *𝑓*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑠,𝑑*

*𝑠,𝑑*

)(1 − *𝑥* )*𝛽*

3Γ ⋅ *𝑥*

+ ∑ *𝐶* − ∑ *𝑓* (1 − *𝑥* )*𝜉*

*𝑠,𝑑*

*𝑠*∈*𝑆*

*𝑠*

*𝑑*∈*𝐷*

*𝑠,𝑑*

*𝑠*

*𝑠*

contradicting our conclusion above that the term has value 1.

( ) *𝑣*

– (4Γ − 2) +

*𝑢*∶(*𝑣,𝑢*)∈*𝐸*

*𝑥𝑢*

≤ 3Γ − (4Γ − 2) + Γ− 1 − *𝑥𝑤*

≤ 1 − *𝑥𝑤*

*<* 1*,*

+ ∑ ∑ *𝑓𝑠,𝑑 𝑥𝑠 𝑦𝑑* − ∑ ∑ *𝑓𝑠,𝑑 𝑥𝑠*

*𝑠*

Note that the transformation of an instance of the clique problem to

to (∗)

# end

*𝜌*∗ ∶= min

scenario *𝑥*

**until** *𝜌*∗ = *𝑡*∗

**return** last *𝑓* ∗

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝜈*∗(*𝑥*) − *𝑓𝑠,𝑑 𝑥𝑠*

∑ ∑

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

the corresponding instance of the maximization problem and its trans- formation 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*](#_bookmark6)*) is* 𝖭𝖯*-hard.*

**Proof.** The instances we used in the proof of the 𝖭𝖯-hardness of the

clique problem. Given is an undirected graph *𝐺* = (*𝑉 , 𝐸*). The question **Proof of Theorem 1.** We use a reduction from the well-known 𝖭𝖯-hard

modeled, setting Γ ≔ *k* as is, whether there exists a clique of size *k* in *G*. We claim that this can be

max ∑ (3Γ ⋅ *𝑥𝑣* − (4Γ − 2) + ∑ *𝑥𝑢* )+ *.*

∑

0≤*𝑥*≤1∶

*𝑣*∈*𝑉 𝑥𝑣* ≤Γ *𝑣*∈*𝑉*

*𝑢*∶(*𝑣,𝑢*)∈*𝐸*

There exist corresponding values for the capacities *c* and demands *b*

modeling the above problem in the form of [(4)](#_bookmark18), since we can assume

w.l.o.g. that deg(*𝑣*) ≥ Γ− 2 holds for all nodes *v* ∈ *V*, as otherwise the node *v* could not be part of any clique of size Γ = *𝑘,* leading to a reduction

by removing this node.

value ≥ Γ if and only if there is a clique of size Γ. Suppose that *C⊆V* is We now show that the solution of the above problem has objective a clique of size Γ in *G*. Then, setting

∶= {

Evaluation Problem have the property that their evaluation value is in- dependent 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* ∈ 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. □

complete ([Downey and Fellows, 1999](#_bookmark49)). Thus, since the parameter Γ It is well-known that the clique problem is W[1]-

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.

*𝑥𝑣*

1*, 𝑣* ∈ *𝐶,*

0*, 𝑣* ∉ *𝐶,*

# Algorithms

yields a solution value of Γ, since we then have for every *v* ∈ *C*, i.e.

= Γ

In this section, we describe three algorithms for the Robust Flow

|*𝐶*|

3Γ ⋅ *𝑥𝑣* − (4Γ − 2) +

times:

∑

*𝑢*∶(*𝑣,𝑢*)∈*𝐸*

*𝑥𝑢* = 3Γ − (4Γ − 2) + Γ− 1 = 1*.*

problem with adaptive mitigation, in increasing order of sophistication.

value at least Γ. Since Σ*v* ∈ *Vxv* ≤ Γ holds by assumption, a requirement Suppose *x* is a solution to the maximization problem with objective

for

∑

* 1. *Outer Linearization Algorithm*

We start with presenting the conceptually most simple algorithm for solving an Adjustable Robust Optimization problem. The *Outer Lin-*

3Γ ⋅ *𝑥𝑣*

– (4Γ − 2) +

*𝑢*∶(*𝑣,𝑢*)∈*𝐸*

?

*𝑥𝑢 >* 0

*earization Algorithm* is based on including the mitigation for all scenarios as variables in a LP. The objective function is represented as the maxi-

*⏟⏞⏞⏞⏟⏞⏞⏞⏟*

≤Γ

mum of the unmitigated shortages over each scenario.

max *𝑡*

s.t. nominal flow constraints (1) for *𝑓*

*𝑠,𝑑*

*𝑠 𝑆*

*𝑑 𝐷*

*𝐶𝑠* −

*𝑑*∈*𝐷 𝑓𝑠,𝑑* (1 − *𝑥𝑠*) for all *𝑠* ∈ *𝑆* and *𝑥* ∈ *𝑋*

*𝑡* ≤ ∑ ∈ ∑ ∈

*𝑑 𝐷*

*𝑓* ′(*𝑥*) ≤ (*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* )(1 − *𝑥𝑠*) for all *𝑠* ∈ *𝑆, 𝑑* ∈ *𝐷* and *𝑥* ∈ *𝑋*

*𝑠 𝑆*

*𝑓* ′(*𝑥*) − ∑ ∈ ∑ ∈

*𝑥𝑠𝑓𝑠,𝑑* for all*𝑥* ∈ *𝑋*

*𝑠,𝑑*

∑

*𝑑*∈*𝐷 𝑓𝑠,𝑑* ≤

′(*𝑥*) ( ∑ )

∑ ∈ *𝑓* ′(*𝑥*) ≤ ∑ ∈

(5)

*𝑠 𝑆*

*𝑠,𝑑*

*𝑠 𝑆*

*𝑓* ′(*𝑥*) ≥ 0 for all *𝑠* ∈ *𝑆, 𝑑* ∈ *𝐷* and *𝑥* ∈ *𝑋*

*𝑠,𝑑*

*𝑥𝑠* ⋅ *𝑓𝑠,𝑑* for all*𝑑* ∈ *𝐷*and*𝑥* ∈ *𝑋*

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

* 1. *Scenario Enumeration Algorithm*

In this section, we describe the Scenario Enumeration Algorithm. In- stead 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 robust- ness. 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*

We use techniques originating from Benders’ decomposition ([Benders, 1962](#_bookmark40)) and linearization with McCormick envelopes ([McCormick, 1976](#_bookmark59)). This general technique has also been applied to several tri-level optimization problems, e.g. by [Alderson et al. (2011)](#_bookmark69); [Sadeghi et al. (2017)](#_bookmark64); [Smith et al. (2007)](#_bookmark68), although the specific formulations vary with different models.

Initialize (∗) to be the following LP: **Algorithm 2:** Iterative Cut Generation Algorithm

max *𝑡*

s.t. nominal flow constraints (1) for *𝑓*

*𝑡* ≤ 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

* + - a supersource *s*0, with arcs (*s*0, *s*), *s* ∈ *S* each having capacity

∑ ∑(

– *𝑓*

)*𝑧*∗

+ ∑ (

*𝐶* −

∑ *𝑓*

)*𝜇*∗

*𝑐*′

= (*𝐶* −

∑ *𝑓*

)(1 − *𝑥* )

*𝑡* ≤

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑐𝑠,𝑑*

*𝑠,𝑑*

*𝑠,𝑑*

*𝑠*∈*𝑆*

*𝑠*

*𝑑*∈*𝐷*

*𝑠,𝑑 𝑠*

*𝑠*0 *,𝑠*

*𝑠*

*𝑑*∈*𝐷*

*𝑠,𝑑 𝑠*

+ ∑ ∑

*𝑓𝑠,𝑑 𝜋*∗

– ∑ ∑

*𝑓𝑠,𝑑 𝑥*∗

* + - arcs from every supplier *s* ∈ *S* to every demand node *d* ∈ *D* with

*𝑠,𝑑*

*𝑠*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

capacity *𝑐*′

= (1 − *𝑥𝑠*)(*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* )

to (∗)

′ **until** *𝜌*∗ = *𝑡*∗;

* + - a supersink *d*0 with arcs (*d, d*0), *d* ∈ *D*, each having capacity *𝑐𝑑,𝑑*0 =

*𝑠,𝑑*

∑*𝑠*∈*𝑆 𝑥𝑠𝑓𝑠,𝑑*

**return** last *𝑓* ∗

**Theorem 2** (Correctness of Alg. 1)**.** *Let f*∗ *be the output of Algorithm* [*1*](#_bookmark19)*. 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](#_bookmark22). The capacity of

this cut provides an upper bound to the total amount of flow in the case

We start with dualizing the model for the mitigation flow [(2)](#_bookmark11). Given some nominal flow *f* and scenario *x*, this results in the following linear program.

min ∑*𝑠*∈*𝑆* ∑( *𝑑*∈*𝐷* (∑*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* ))(1 − *𝑥𝑠*)*𝛽𝑠,𝑑* ∑ ∑

+

*𝑠*∈*𝑆*

*𝐶𝑠* −

*𝑑*∈*𝐷 𝑓𝑠,𝑑* (1 − *𝑥𝑠*)*𝜉𝑠* +

*𝑑*∈*𝐷*

*𝑠*∈*𝑆 𝑓𝑠,𝑑 𝑥𝑠𝑦𝑑*

(6)

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.

lution *t*∗ that is obtained always overestimates the robustness value of an Since these constraints on *t* represent all valid upper bounds, the so- optimal solution. The value *𝜌*∗ gives the robustness value of the current

flow *f*. Thus, always *𝜌*∗ ≤ *t*∗ holds. If *𝜌*∗ = *𝑡*∗ then the current solution

s.t. *𝜉𝑠* + *𝛽𝑠,𝑑* + *𝑦𝑑* ≥ 1 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

*𝜉𝑠, 𝛽𝑠,𝑑 , 𝑦𝑑* ≥ 0 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

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* )*𝛽 ,* (1 − *𝑥* )*𝜉* and *x y*

has a robustness value equal to the upper bound *t*∗, which implies that

*𝑠 𝑠,𝑑*

*𝑠 𝑠 s d*

it must be optimal. □

* 1. *Cut-generation algorithms for the robust flow problem*

that appear in the objective function. Since we require that *xs* ∈ {0, 1}, we can replace *xsyd* by the variable *𝜋s,d* with the constraints *𝜋𝑠,𝑑* ≥ *𝑥𝑠* +

*𝑦𝑑* − 1 and *𝜋s,d* ≥ 0. Since the coeﬃcient in front of *xsyd* is always pos-

itive, in an optimal solution we will choose *𝜋𝑠,𝑑* = max{*𝑥𝑠* + *𝑦𝑑* − 1*,* 0}. Since *xs* ∈ {0, 1} and *yd* ∈ [0, 1], we then have

We now describe how a cut generation algorithm can solve the Ro- bust Flow problem. The main difference to the other presented algo-

*𝜋𝑠,𝑑*

= {0*, 𝑥𝑠*

= 0*,*

= *𝑥𝑠* ⋅ *𝑦𝑑 .*

rithms is that we allow here an implicitly represented set of scenarios

*X*(*A*, Γ) according to [Definition 3](#_bookmark9). This makes the other algorithms im-

practical, since they require to enumerate all scenarios. Random sam-

pling from the scenario set is also not suﬃcient, since we want an exact solution. Thus we use a cutting plane approach.

*𝑦𝑑 , 𝑥𝑠* = 1*,*

Similarly, we can replace (1 − *𝑥𝑠*)*𝛽𝑠,𝑑* with the variable *zs,d* and the con- straints *𝑧𝑠,𝑑* ≥ *𝛽𝑠,𝑑* − *𝑥𝑠* and *zs,d* ≥ 0. Analogously, the term (1 − *𝑥𝑠*)*𝜉𝑠* can be replaced with the variable *𝜇s* and the constraints *𝜇𝑠* ≥ *𝜉𝑠* − *𝑥𝑠* and

*𝜇s* ≥ 0.

This leads to the following IP: by [Schlenker (1989)](#_bookmark66). [Table 1](#_bookmark25) illustrates the chosen parameter values for

min ∑ ∑

∑ ∑ ∑ ∑

*𝑠*∈*𝑆*

*𝑑*∈*𝐷*

*𝑠,𝑑*

(*𝑐*

– *𝑓*

)*𝑧* + ∑

(*𝐶* − ∑

*𝑓* )*𝜇*

NETGEN. This ensures that we get bipartite graphs.

NETGEN generates capacitated network flow problems. Since the

+ *𝑠*∈*𝑆 𝑑*∈*𝐷 𝑓𝑠,𝑑 𝜋𝑠,𝑑* − *𝑠*∈*𝑆 𝑑*∈*𝐷 𝑓𝑠,𝑑 𝑥𝑠*

*𝑠,𝑑*

*𝑠,𝑑*

*𝑠*∈*𝑆*

*𝑠*

*𝑑*∈*𝐷*

s.t. *𝜉𝑠* + *𝛽𝑠,𝑑* + *𝑦𝑑* ≥ 1 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

*𝑠,𝑑*

*𝑠*

*𝐴𝑥* ≤ Γ

*𝑧𝑠,𝑑* ≥ *𝛽𝑠,𝑑* − *𝑥𝑠* for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

*𝜇𝑠* ≥ *𝜉𝑠* − *𝑥𝑠* for all *𝑠* ∈ *𝑆*

*𝜋𝑠,𝑑* ≥ *𝑥𝑠* + *𝑦𝑑* − 1 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

*𝜉𝑠 , 𝛽𝑠,𝑑 , 𝑦𝑑 , 𝑧𝑠,𝑑 , 𝜇𝑠 , 𝜋𝑠,𝑑* ≥ 0 for all *𝑠* ∈ *𝑆* and *𝑑* ∈ *𝐷*

*𝑥𝑠* ∈ {0*,* 1} for all *𝑠* ∈ *𝑆*

(7)

generated total capacity of the source nodes is equal to the total de-

we multiply each source capacity by a common parameter *𝜂 >* 1 to yield mand, no residual capacity is available in the generated networks. Thus, some residual capacity for mitigation. We used the value *𝜂* = 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 sig- nificantly on the robustness value. This is illustrated in [Fig. 2](#_bookmark26), showing

The idea of our algorithm is based on evaluating [(7)](#_bookmark23) 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)](#_bookmark23).

**Theorem 3** (Correctness of Algorithm 2)**.** *Let f*∗ *be the output of*

*than f*∗*.* [*Algorithm 2*](#_bookmark21)*. Then, there does not exist a feasible flow f which is more robust*

**Proof.** Let (*z*∗, *𝜇*∗, *𝜋*∗, *x*∗) be a feasible solution to [(7)](#_bookmark23). Note that fea-

sibility is independent of the flow *f*. Thus, since [(7)](#_bookmark23) is a minimization

problem, the term

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)](#_bookmark8), for a

to more robust solutions with less shortages. The parameter Γ of failing better overview. This means that lower values are better and correspond

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

∑ ∑(*𝑐𝑠,𝑑* − *𝑓𝑠,𝑑* )*𝑧*∗

+ ∑ (*𝐶𝑠* −

∑ *𝑓𝑠,𝑑*

)*𝜇*∗

ness value, particularly if the parameter Γ is not too big, i.e. Γ ≤ 3. For that taking mitigations into account provides a benefit in the robust-

*𝑠,𝑑 𝑠*

∈ ∈ ∈ ∈

+ ∑ ∑ *𝑓𝑠,𝑑 𝜋*

– ∑ ∑ *𝑓𝑠,𝑑 𝑥*

*𝑠 𝑆 𝑑 𝐷*

*𝑠 𝑆*

*𝑑 𝐷*

larger values of Γ the scenarios become so restricting, that also mitiga-

∗

*𝑠,𝑑*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

∗

*𝑠*

*𝑠*∈*𝑆 𝑑*∈*𝐷*

tion 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

to (∗) only constraints of this form (for feasible (*z*∗, *𝜇*∗, *𝜋*∗, *x*∗)), the is an upper bound on the optimal objective value of [(7)](#_bookmark23). Since we add objective value *t*∗ of (∗) in the last iteration is an upper bound on the

when *𝜌*∗ ≥ *t*∗. Thus, *f*∗ has a robustness value of *𝜌*∗ which is at least as optimal robustness of any feasible flow. [Algorithm 2](#_bookmark21) terminates only

large as the upper bound for the robustness of any feasible flow. Hence,

*f*∗ has optimal robustness. □

The finite convergence of [Algorithm 2](#_bookmark21) can be seen as follows: Each solution of the IP [(7)](#_bookmark23) 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 termi- nates.

The advantage of the Scenario Enumeration [Algorithm 1](#_bookmark19) compared to the Iterative Cut Generation [Algorithm 2](#_bookmark21) 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).](#_bookmark63)

# 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 Optimiza- tion, LLC, 2019). The max-flow problems in the Scenario Enumeration](#_bookmark52) algorithm are solved with NetworkX ([Hagberg et al., 2008](#_bookmark53)). The bench- marks 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](#_bookmark48)), on which this paper is based, for additional numerical studies.

We generate instances of our Robust Flow problem with NETGEN ([Klingman et al., 1974](#_bookmark57)) which is widely used throughout the literature to evaluate network flow algorithms. We are using an implementation

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

In [Fig. 3](#_bookmark27), we show the influence of the parameter *𝜂* controlling the diﬃcult that the choice of a flow does not impact the robustness value.

instance. As can be seen, for small values of *𝜂 <* 1.1, there is no big capacity of the suppliers on the obtained objective value for a network

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

(*𝜂* ≥ 1.2), the difference between the two optimization models becomes

larger. Larger over-capacities than *𝜂* = 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

*𝜂* = 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](#_bookmark28)(a) shows the influence of the number of suppliers |*S*|. Since the Outer Linearization and the Scenario Enumeration Algorithm both it- erate over all scenarios, their running times increase at least linearly with the number of scenarios. In contrast, the Iterative Cut Gener- ation 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](#_bookmark28)(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*

1 *𝑆* ⋅ *𝐷*

105 100% 3⋅105

| | | |



2 | | | |

|*𝑆*|⋅|*𝐷*|

the number of failing suppliers Γ. The chosen **Fig. 2.** Dependence of the objective value on

= 30 suppliers. The total

| |

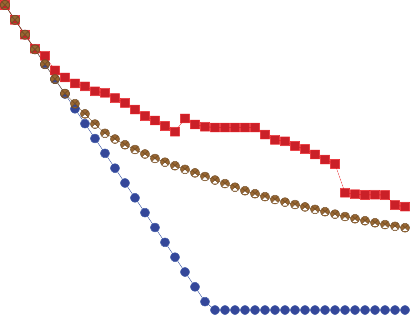
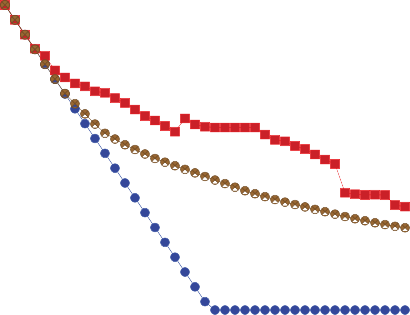
instance contains *𝑆*

supply is 105.





the parameter *𝜂* specifying the over-capacity **Fig. 3.** Dependence of the objective value on



of the suppliers. The chosen instance contains

= 30 suppliers, the number of failing sup-

| |

*𝑆*

pliers was set to Γ = 3. The total supply is 105.





* In [Fig. 4](#_bookmark28)(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 𝑆* . As expected, the running time of the Outer Linearization

*𝛤*

and the Scenario Enumeration Algorithm increases since it is at least

Iterative Cut Generation decreases with increasing Γ, since the found linear in the number of scenarios. However, the running time of the

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 algo- rithm. However, for very small scenario sets, the Outer Linearization Algorithm is faster. For large scenario sets, the Iterative Cut Gener- ation Algorithm is especially suited, since the running time does *not* increase when the number of suppliers considered in a failure sce- nario increases.
* The Scenario Enumeration Algorithm is generally the slowest one. However, the algorithm is more flexible than the others, since it al-

lows 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 bet- ter than the other algorithms, we can apply it to significantly larger in- stances. The following plots illustrate its running time behavior on larger instances. [Fig. 5](#_bookmark30)(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 Γ ≥ 4 failing

suppliers.

[Fig. 6](#_bookmark31) illustrates the behavior of the Iterative Cut Generation Algo- rithm 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 opti- mality of this solution.







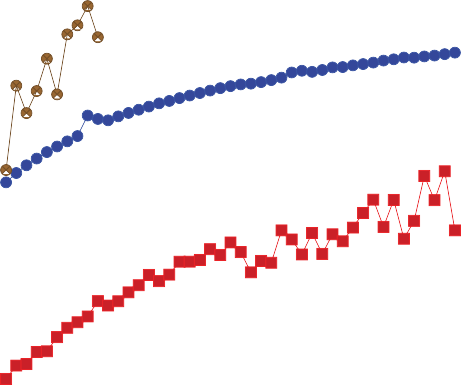
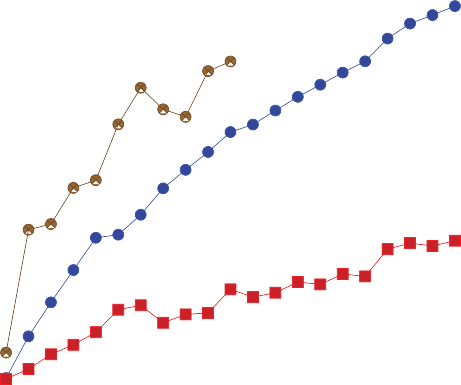
**Fig. 4.** Comparison of the running time of the three algorithms, depending on multiple parameters of the instances. The default parameters used here are *𝑆*

| |

= 15*,* Γ = 3.

= 15*,*

|*𝐷*|



# 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](#_bookmark21) 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 capaci- ties 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 net-

work 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 in- stalled by opening some depots out of a given set of possible lo- cations. 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.









= 30*, 𝐷* = 30*,*



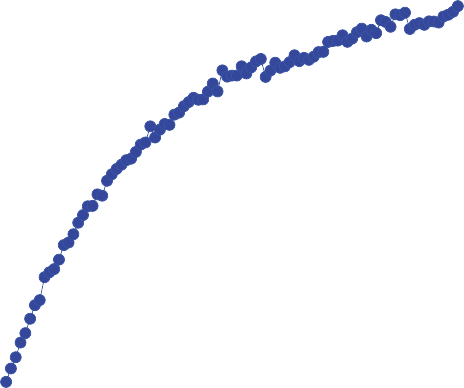
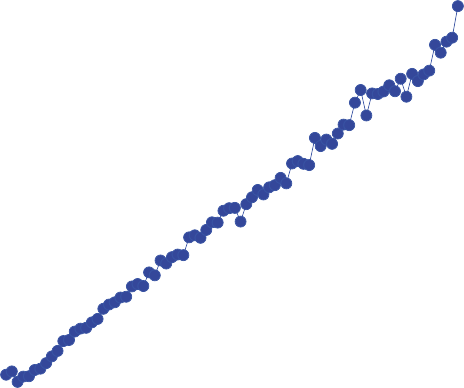
| |

| |

**Fig. 5.** Dependence of the running time of the Iterative Cut Generation Algorithm on various parameters. The default parameters used here are: *𝑆*

Γ = 10.

Our work shows that it is possible to take second stage mitiga- tion already into account when optimizing supply chains. Our ap- proach is able to handle large scenario sets by using an implicit de- scription of the scenario set. The approach offers great flexibility, since many additional constraints can be modeled into the LP [(1)](#_bookmark7) with- out extensive modifications to the algorithm. The results also indi- cate 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 Algo- rithm 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 sig- nificantly improved its performance.

**Fig. 6.** The evolution of the robustness of the current solutions over the iterations of the It- erative Cut Generation Algorithm. Note that a robustness value closer to 0 is better. Here, the run of the Iterative Cut Generation Algo- rithm for a randomly generated, medium-sized



= 30*, 𝐷* = 30 and Γ = 4 is

instance with |*𝑆*| | |

shown. The upper bound corresponds to the objective value of the outer LP. Also the best found robustness value, i.e. the evaluated ro- bustness 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 vec-

tor *𝑥* = (*𝑥𝑠* )*𝑠*∈*𝑆* representing the scenario of failing suppliers, i.e. there

should exist a matrix *A* and a vector *d* such that

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 short- age scenarios. We will use this information, to show that an aﬃne deci- sion rule is not optimal in this case.

* In the scenario (*s*1 , *s*3 ) the unique optimal mitigation is given by

*𝑓* ′ = *𝐴𝑥* + *𝑑*

*𝑓* ′

*𝑠*2 *,𝑑*3

*, 𝑓* ′

*𝑠*3 *,𝑑*1

= 1

= 1*,*

holds for every scenario *x* ∈ *X*.

Consider the network depicted in [Fig. A1](#_bookmark32) with suppliers *𝑆* =

{*𝑠*1 *,* … *, 𝑠*4 } and demand nodes *𝐷* = {*𝑑*1 *,* … *, 𝑑*4 }. We use the scenario set

and all other entries of *f*′ set to 0.

* In scenario (*s*2, *s*4) the nonzero entries of the unique optimal mitiga- tion are given by:

*𝑋* = *𝑋*(2) given as the Γ-robust scenario set with Γ = 2.

Consider now the failure scenario (*s* , *s* ). Then, a shortage of 1 each

*𝑓* ′

*𝑠*1 *,𝑑*2

*, 𝑓* ′

*𝑠*3 *,𝑑*4

= 1

= 1*.*

2 3

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

* In scenario (*s*1 , *s*4 ) it is given by:

*𝑓* ′ = 1*, 𝑓* ′ = 1*.*

shortage at *d* . Hence, in an optimal mitigation *f*′, we have *𝑓* ′

= 1.

*𝑠*2 *,𝑑*4

*𝑠*3 *,𝑑*4

2 *𝑠*4 *,𝑑*2

Thus, there only remains shortage at *d*3, which can be mitigated by *d*1

Suppose that an aﬃne decision rule yields an optimal solution for this example. Then, since the optimal mitigation is always unique for

by setting *𝑓* ′

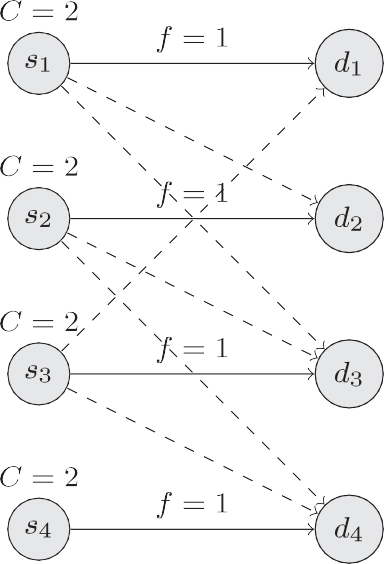
*𝑠*1 *,𝑑*3

= 1. The other entries of the mitigation vector *f*′ are 0.

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* ∶ ℝ4 → ℝ4×4*, 𝑥* ↦ *𝑓* ′

maps a vector *𝑥* = (*𝑥𝑠* )*𝑠*∈*𝑆* indicating the failure scenario to a vector *𝑓* ’ =



(*𝑓* ’ )

representing the mitigation flow.

*𝑠,𝑑 𝑠*∈*𝑆,𝑑*∈*𝐷*

By writing a scenario *x* as

*𝑥𝑠*1

⎛ ⎞

*𝑥𝑠*2

*𝑥* = ⎜*𝑥* ⎟*,*

*𝑠*3

*𝑥𝑠*4

⎜⎝ ⎟⎠

we have the following equality since *h* is aﬃne:

⎛0⎞

1 0 1 0

⎛1⎞

⎛0⎞

⎛1⎞

1

*ℎ*⎜ ⎟ = *ℎ*⎜ ⎟ + *ℎ*⎜ ⎟ − *ℎ*⎜ ⎟*.* (A.1)

⎜⎝0⎟⎠

1

⎜⎝0⎟⎠

0

⎜⎝1⎟⎠

0

⎜⎝1⎟⎠

Suppose now that the function *h* would always yield the optimal miti- gation in each scenario. Then, the following must hold by our analysis of the optimal mitigation in the scenarios considered above, using the

notation *𝑒𝑠𝑖 ,𝑑𝑗* for the unit vector of the component (*si, dj* ):

**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 *si* is connected

to *di* , *𝑑𝑖*+1 and *𝑑𝑖*+2 *,* where indices are taken modulo 4.

11

⎛0⎞

⎜⎝0⎟⎠

*ℎ*⎜1⎟ = *𝑒*

1

*𝑠*1 *,𝑑*3

+ *𝑒*

*𝑠*4 *,𝑑*2

⎛1⎞

⎜⎝0⎟⎠

*, ℎ*⎜1⎟ = *𝑒*

0

*𝑠*2 *,𝑑*3

+ *𝑒*

*𝑠*3 *,𝑑*1 *,*

⎛0⎞

*ℎ*⎜0⎟ =

1

⎜⎝1⎟⎠

*𝑒𝑠*1 *,𝑑*2 +

⎛1⎞

⎜⎝1⎟⎠

*𝑒𝑠*3 *,𝑑*4 *, ℎ*⎜0⎟ =

0

*𝑒𝑠*2 *,𝑑*4

+ *𝑒𝑠*3 *,𝑑*1 *.*

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](https://doi.org/10.1287/trsc.2017.0749).

Alvarez-Miranda, E., Ljubić, I., Raghavan, S., Toth, P., 2014. The recoverable ro- bust two-level network design problem. INFORMS J. Comput. 27 (1), 1–19. doi:[10.1287/ijoc.2014.0606](https://doi.org/10.1287/ijoc.2014.0606).

Aneja, Y.P., Chandrasekaran, R., Nair, K.P.K., 2001. Maximizing residual flow under an

By plugging this in, we see directly that this contradicts the aﬃnity condition [(A.1)](#_bookmark33). 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 pro- vides. Consider the failure of a supplier *si* . Then, a shortage is created at demand node *di* . The mitigation at demand node *di* can be done from

the suppliers *𝑠𝑖*−2 and *𝑠𝑖*−1 *,* where indices are taken modulo 4. Thus, the

optimal aﬃne function *h* should fulfill

*ℎ*(*𝑒𝑠* ) = *𝛼* ⋅ *𝑒𝑠 ,𝑑* + *𝛼* ⋅ *𝑒𝑠 ,𝑑 ,*

*𝑖 𝑖*−2 *𝑖 𝑖*−1 *𝑖*

arc destruction. Networks 38 (4), 194–198. doi:[10.1002/net.10001](https://doi.org/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](https://doi.org/10.1287/opre.1070.0428).

[Ben-Tal,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0007) [A., El Ghaoui, L., Nemirovski, A., 2009. Robust Optimization, 28. Princeton Uni- versity Press.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0007)

Ben-Tal, A., Golany, B., Nemirovski, A., Vial, J.-P., 2005. Retailer-supplier flexible com- mitments contracts: a robust optimization approach. Manuf. Serv. Oper. Manag. 7 (3), 248–271. doi:[10.1287/msom.1050.0081](https://doi.org/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](https://doi.org/10.1007/BF01386316).

Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2004. Adjustable ro- bust solutions of uncertain linear programs. Math. Program. 99 (2), 351–376. doi:[10.1007/s10107-003-0454-y](https://doi.org/10.1007/s10107-003-0454-y).

Bertsimas, D., Litvinov, E., Sun, X.A., Zhao, J., Zheng, T., 2013. Adaptive robust optimiza-

for some *𝛼* ∈ ℝ≥0

. The choice of the same factor *𝛼* in front of *𝑒*

*𝑠𝑖*−2 *,𝑑𝑖*

and

tion for the security constrained unit commitment problem. IEEE Trans. Power Syst. 28 (1), 52–63. doi:[10.1109/TPWRS.2012.2205021](https://doi.org/10.1109/TPWRS.2012.2205021).

*𝑒𝑠𝑖*−1 *,𝑑𝑖* follows from the symmetry of the network: Locally, with respect

to single suppliers, the nodes *𝑠𝑖*−2 and *𝑠𝑖*−1 are equivalent.

Similarly by symmetry, we can assume that the value of *𝛼* is the same for every supplier *si* . 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*(*𝑒𝑠*1 + *𝑒𝑠*2 ) = *ℎ*(*𝑒𝑠*1 ) + *ℎ*(*𝑒𝑠*2 ) = *𝛼𝑒𝑠*3 *,𝑑*1 + *𝛼𝑒𝑠*4 *,𝑑*1 + *𝛼𝑒𝑠*4 *,𝑑*2 + *𝛼𝑒𝑠*1 *,𝑑*2 *,*

should be feasible. By the capacity requirement of supplier *s*4, the fol- lowing must hold for feasibility:

*𝑓**𝑠*4 *,𝑑*4 + 2*𝛼* ≤ *𝐶𝑠*4 *,*

i.e.

1 + 2*𝛼* ≤ 2 ⟺ *𝛼* ≤ 1 *.*

4

Thus, the optimal choice is *𝛼* = 1 . The corresponding mitigated amount in each scenario is then 4 = 4 4 1 = 1. Thus, since the shortage was 2,

*𝛼* ⋅ 4

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 func- tion in the scenario *x* such that a desired value is met for every scenario

*x* ∈ *X* contained in the scenario set. In particular, this is the case if the

mally, we can find a matrix *A* and vector *d* such that for every *x* ∈ *X* we scenario set consists only of scenarios with one failing supplier, For-

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* ∈ *X* is linearly independent by the given as- sumption.

Hence, if the family of scenarios (*x*)*x* ∈ *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*| ≤ |*S*| in the number of suppliers. Thus, an approach which explicitly constructs an optimal mitigation for each scenario also leads to an eﬃciently computed opti- mal solution.

# References

[Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 2014. Network Flows: Theory, Algorithms and Applications. Pearson.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0001)

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](https://doi.org/10.1287/ics.2011.0047).

Bertsimas, D., Nasrabadi, E., Stiller, S., 2013. Robust and adaptive network flows. Oper.

Res. 61 (5), 1218–1242. doi:[10.1287/opre.2013.1200](https://doi.org/10.1287/opre.2013.1200).

[Birge, J.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0013), [Louveaux, F., 2011. Introduction to Stochastic Programming. Springer](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0013).

Brown, G., Carlyle, M., Salmerón, J., Wood, K., 2006. Defending critical infrastructure.

Interfaces 36 (6), 530–544. doi:[10.1287/inte.1060.0252](https://doi.org/10.1287/inte.1060.0252).

Brown, G.G., Carlyle, W.M., Wood, R.K., 2008. Department of Homeland Secu- rity Bioterrorism Risk Assessment. National Academies Press, Washington, D.C. doi:[10.17226/12206](https://doi.org/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](https://doi.org/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](https://doi.org/10.1287/trsc.1100.0340).

[Diessel,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0018) [E., 2019. Risk aware flow optimization and application to logistics networks, Master Thesis Technische Universität Kaiserslautern, Department of Mathematics.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0018)

[Downey,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0019) [R.G., Fellows, M.R., 1999. Parameterized Complexity. Monographs in Computer Science. Springer.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0019)

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](https://doi.org/10.1002/net.20188).

Garg, M., Smith, J.C., 2008. Models and algorithms for the design of survivable multi- commodity flow networks with general failure scenarios. Omega 36 (6), 1057–1071. doi:[10.1016/j.omega.2006.05.006](https://doi.org/10.1016/j.omega.2006.05.006).

Gurobi Optimization, LLC, 2019. Gurobi Optimizer Reference Manual.

[Hagberg,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0022) [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.), Pro- ceedings of the Seventh Python in Science Conference, pp. 11–15.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0022)

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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1287/mnsc.20.5.814).

Lozano, L., Smith, J.C., 2017. A backward sampling framework for interdic- tion problems with fortification. INFORMS J. Comput. 29 (1), 123–139. doi:[10.1287/ijoc.2016.0721](https://doi.org/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](https://doi.org/10.1007/s10107-017-1166-z).

McCormick, G.P., 1976. Computability of global solutions to factorable nonconvex pro- grams: part I — convex underestimating problems. Math. Program. 10 (1), 147–175. doi:[10.1007/BF01580665](https://doi.org/10.1007/BF01580665).

[Phillips,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0029) [C.A., 1993. The network inhibition problem. In: Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing, pp. 776–785.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0029)

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](https://doi.org/10.1002/nav.21541).

[Roberto,](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0031) [M.A., Levesque, L.C., 2004. Managing risk to avoid supply-chain breakdown.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0031)

[MIT Sloan Manag. Rev. 46 (1), 53–60.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0031)

Rockafellar, R.T., Uryasev, S., 2000. Optimization of conditional value-at-risk. J. Risk 2, 21–41. doi:[10.21314/JOR.2000.038](https://doi.org/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](https://doi.org/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.](http://www.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](https://doi.org/10.1111/poms.12887).

Simchi-Levi, D., Wang, H., Wei, Y., 2019. Constraint generation for two-stage robust net- work flow problems. INFORMS J. Optim. 1 (1), 49–70. doi:[10.1287/ijoo.2018.0003](https://doi.org/10.1287/ijoo.2018.0003). Smith, J.C., Lim, C., Sudargho, F., 2007. Survivable network design under op- timal and heuristic interdiction scenarios. J. Glob. Optim. 38 (2), 181–199.

doi:[10.1007/s10898-006-9067-3](https://doi.org/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.](http://refhub.elsevier.com/S2192-4406(20)30002-2/sbref0037)

Wood, R.K., 1993. Deterministic network interdiction. Math. Comput. Model. 17 (2), 1–

18. doi:[10.1016/0895-7177(93)90236-R](https://doi.org/10.1016/0895-7177(93)90236-R).

Yanikoğlu, I., Gorissen, B.L., den Hertog, D., 2018. A survey of adjustable robust optimiza- tion. Eur. J. Oper. Res. 277 (3), 799–813. doi:[10.1016/j.ejor.2018.08.031](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1287/opre.2017.1714).