

# Chapter 11

## Robust Network Design



Arie M. C. A. Koster and Daniel R. Schmidt

### 1 Introduction

Designing a network is usually done on the basis of a forecast of the future/expected demand. Such a forecast will by definition not be an accurate representation of the reality. If the design process involves long-term and/or strategic decisions, the quality of the forecast determines the feasibility of the network design for its future purpose. Where an increase of the forecast values might be a simple solution to this problem, it is obvious that such a line of action might be too rough and hence unnecessary costly. Robust optimization offers a more informed alternative in such cases.

In the following, we first introduce the basics of robust optimization. Next, we survey its application to single- and multicommodity network design. At appropriate times, extensions of the basic idea of robust optimization are also introduced.

---

A. M. C. A. Koster (✉)  
Lehrstuhl II für Mathematik, RWTH Aachen University, Aachen, Germany  
e-mail: [koster@math2.rwth-aachen.de](mailto:koster@math2.rwth-aachen.de)

D. R. Schmidt  
Institute for Computer Science V, University of Bonn, Bonn, Germany  
e-mail: [daniel.schmidt@uni-bonn.de](mailto:daniel.schmidt@uni-bonn.de)

## 2 Robust Optimization

### 2.1 What Is Robust Optimization?

According to a text book by Ben-Tal et al. (2009), robust optimization is a “*methodology for handling optimization problems with uncertain data*”. We extend this notion of robustness and say that a solution to an optimization problem is robust if it is feasible for a prescribed range of scenarios rather than in a single situation. Let us illustrate this concept with an example. Suppose we are to plan a network for an internet provider. We are provided with forecasts for the planning period and as a first step, we convert the forecasts into hard numbers for our problem input. This step will almost certainly introduce rounding errors and is prone to rule out potentially useful solutions. After this “rounding” step, we run an *exact* optimization algorithm – but only *after* we introduced errors and inaccuracies! How can we make sure that this solution is even feasible for the real (unknown) network requirements? There is another problem with our approach. We might be planning a network that sees different usage scenarios throughout the day, and while classical optimization can find a network for each scenario, it lacks the methodology to find a network that works in *all* of them. The methodology in this chapter gives us the ability to find solutions that are robust against imprecisions in the input and shifting use cases. Among other things, it will let us cope with imprecise numbers and lets us plan a network that can support different traffic peaks without requiring that all peaks can be handled simultaneously. Throughout, our approach will be as follows. First, we need to identify a set of possible network configurations or scenarios. This set is called the *uncertainty set*. Then, we will look for *worst-case* robust solutions, i.e. solutions that are feasible no matter which scenario occurs. The challenge here is to carefully select an appropriate uncertainty set: The broader the set, the more expensive our solution becomes. Still, if we were to accept that some solution is *not* feasible in all scenarios, we would accept that in some scenarios we violate our side constraints. Thus, a worst-case model is appropriate if in the application, safety is critical and a failure of the optimized system is not permitted or more expensive than guarding against it. If we are sure that all parameters realizations will occur eventually or if the probability distribution of the realizations is not known, then worst-case robustness is a good modeling choice as well (if only for the lack of alternatives).

### 2.2 Chance-Constrained Model

Robust optimization can be viewed as a specialization of chance-constrained optimization. As it is often impossible to map all possible inputs onto the uncertainty set, there remains a (hopefully small) chance that a robust solution is infeasible. Stated otherwise, a chance-constrained optimization model can be reformulated as

a robust optimization model by determining an uncertainty set guaranteeing that solutions to the robust optimization model satisfy the original constraints with high probability. More formally, assuming uncertainties in the constraint matrix of an arbitrary linear program  $\min\{c^T x \mid Ax \geq b, x \geq 0\}$  only, a chance-constrained optimization program in its general form is:

$$\text{Minimize} \quad \sum_{j=1}^n c_j x_j \quad (11.1)$$

$$\text{Subject to} \quad \Pr \left( \sum_{j=1}^n a_{ij} x_j \geq b_i \right) \geq 1 - \epsilon_i \quad \forall i = 1, \dots, m \quad (11.2)$$

$$x_j \geq 0 \quad \forall j = 1, \dots, n. \quad (11.3)$$

Here,  $\epsilon_i > 0$  is the maximum probability of violating the  $i$ -th constraint and the matrix entries  $a_{ij}$  are no longer deterministic values, but random variables following a (possibly unknown) distribution. Alternatively, all constraints can be considered jointly in a single chance constraint:

$$\Pr \left( \sum_{j=1}^n a_{ij} x_j \geq b_i \quad \forall i = 1, \dots, m \right) \geq 1 - \epsilon \quad (11.4)$$

In this case, a single value  $\epsilon > 0$  specifies the maximum probability that a chance-constrained solution is infeasible.

Chance-constrained models are often difficult to solve, in particular if information on the probability distribution is not (or only limitedly) available. Tractability is further restricted by dependencies between the random variables. In the context of network design, we refer to Pascali (2009) for a chance-constrained approach. In the context of broadband wireless networks, we are aware of another work reducing the model to a deterministic problem in case of independent random variables, see Claßen et al. (2014). Alternatively, by following a robust optimization approach, we sometimes can guarantee that the solution satisfies the inequalities with high probability, either in theory, or in practice (by evaluating historical data). Therefore, we next describe a number of commonly used uncertainty sets.

### 2.3 Interval Uncertainty

In many cases, parts of the constraint matrix are based on physical measurements or forecasts and are thus not known with arbitrary precision. To capture this kind of uncertain input, assume that each coefficient  $a_{ij}$  of  $A$  has a nominal value  $\bar{a}_{ij}$  (the value that was measured or predicted) and that the true value for  $a_{ij}$  can deviate by at most  $\hat{a}_{ij} \geq 0$  from our nominal choice. We define an uncertainty set  $\mathcal{U}_I$  that

consists of all matrices  $A$  with coefficients  $a_{ij} \in [\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$ :

$$\mathcal{U}_I := \left\{ (a_{ij})_{i,j=1}^{m,n} \in \mathbf{R}^{m \times n} \mid a_{ij} \in [\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}] \forall i, j \right\}. \tag{11.5}$$

Since there is no coupling between the individual coefficients, the worst-case scenario occurs when all coefficients deviate in the worst possible way. This happens when  $a_{ij}$  is set to  $\bar{a}_{ij} - \hat{a}_{ij}$  for all  $i, j$  (assuming a system of type  $Ax \geq b$ ) and results in the following program.

$$\text{Minimize } \sum_{j=1}^n c_j x_j \tag{11.6}$$

$$\text{Subject to } \sum_{j=1}^n (\bar{a}_{ij} - \hat{a}_{ij}) x_j \geq b_i \quad \forall i = 1, \dots, m \tag{11.7}$$

$$x_j \geq 0 \quad \forall j = 1, \dots, n. \tag{11.8}$$

This is a classical result by Soyster (1973). As an example, suppose that our input numbers were measured with an accuracy of 1%. Then, we set  $\bar{a}_{ij}$  to the value that was measured and let  $\hat{a}_{ij} = 0.01 \cdot \bar{a}_{ij}$  for all  $i, j$ .

### 2.4 Budget Uncertainty

In a practical setting, it is unlikely that all coefficients deviate in the worst possible way at the same time. In consequence, solutions from the interval uncertainty model tend to be unnecessarily costly. To obtain a less conservative model, we assume an uncertainty budget that limits the total deviation. Changing the budget will allow us to control the conservatism of the model.

In order to control the level of robustness, let us introduce a parameter vector  $\Gamma \in \mathbf{Z}_{\geq 0}^m$  whose  $i$ -th entry  $\Gamma_i$  decides how many coefficients of the  $i$ -th constraint may deviate from their nominal value at the same time.<sup>1</sup> We define the following set of uncertain matrices based on the choice of  $\Gamma$ .

---

<sup>1</sup>The model can be extended to choose a fractional  $\Gamma \in \mathbf{R}_{\geq 0}^m$  with the interpretation that  $\lfloor \Gamma_i \rfloor$  coefficients of constraint  $i$  deviate maximally and a single coefficient  $a_{ij}$  of constraint  $i$  deviates by the remaining amount  $(\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{ij}$ . The extension is straight-forward, yet omitting it makes the exposition significantly easier. For the full model, see Bertsimas and Sim (2004).

$$\mathcal{U}_B(\Gamma) := \left\{ (a)_{i,j=1}^{m,n} \in \mathbf{R}^{m \times n} \left| \begin{array}{l} S_1, \dots, S_m \subseteq \{1, \dots, n\} \\ |S_i| \leq \Gamma_i \quad \forall i = 1, \dots, m \\ a_{ij} \in [\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}], \quad \text{if } j \in S_i \\ a_{ij} = \bar{a}_{ij}, \quad \text{otherwise} \end{array} \right. \right\}.$$

We say that a solution is  $\Gamma$ -robust if it is feasible for all  $A \in \mathcal{U}_B(\Gamma)$ . By increasing some  $\Gamma_i$ , we increase the robustness as well as the cost of the solution; this is what we call the price of robustness. In this way, solving the model for different values of  $\Gamma$  allows us to find a good trade-off between robustness and cost; the extreme cases being the interval model (set  $\Gamma_i = n$  for all  $i = 1, \dots, m$ ) or a non-robust model (set  $\Gamma_i = 0$  for all  $i = 1, \dots, m$ ). We have the following program with respect to  $\mathcal{U}_B(\Gamma)$ .

$$\text{Minimize} \quad \sum_{j=1}^n c_j x_j \tag{11.9}$$

$$\text{Subject to} \quad \sum_{j=1}^n \bar{a}_{ij} x_j - \max_{\substack{S \subseteq \{1, \dots, n\} \\ |S| \leq \Gamma_i}} \sum_{j \in S} \hat{a}_{ij} x_j \geq b_i \quad \forall i = 1, \dots, m \tag{11.10}$$

$$x_j \geq 0 \quad \forall j = 1, \dots, n \tag{11.11}$$

If we fix a selection  $S$  of deviating coefficients in any constraint  $i$  of (11.9)–(11.11), then this constraint is most restrictive if  $a_{ij}$  deviates to the lower bound  $\bar{a}_{ij} - \hat{a}_{ij}$  for all  $j \in S$ , as  $x_j \geq 0$ . This is modeled by the reformulated constraint (11.10). Program (11.9)–(11.11) can be casted into a linear program by replacing the inner optimization problem by its dual; a method that we shall see in more detail in Sect. 5.3.

The budget uncertainty model has nice theoretical properties with respect to chance-constrained optimization: Let  $\tilde{A}$  result by randomly (but symmetrically) perturbing the coefficients of the original matrix  $A$  while obeying the maximum deviations. Then, the probability that a  $\Gamma$ -robust solution violates the  $i$ -th constraint of  $\tilde{A}x \geq b$  is bounded by  $\exp(-\Gamma_i^2/2n)$ ; independently of the distribution of the perturbation.<sup>2</sup>

---

<sup>2</sup>A tighter, but more involved bound can be shown; see Bertsimas and Sim (2004).

## 2.5 Polyhedral Uncertainty and the Robust Counterpart

To model interval and budget uncertainty, we have collected all possible realizations of the constraint matrix  $A$  in an uncertainty set  $\mathcal{U} \subseteq \mathbf{R}^{m \times n}$  and looked for a solution  $x \geq 0$  that is robust feasible, i.e., one that satisfies  $Ax \geq b$  for all choices of  $A \in \mathcal{U}$ . In general, we can choose any closed, bounded set  $\mathcal{U}$  as our uncertainty set. Furthermore, we can always replace  $\mathcal{U}$  by its convex hull  $\text{conv } \mathcal{U}$  without changing the feasible region of the uncertain program. In fact, we shall assume that  $\mathcal{U}$  is a polytope in the following. Given an arbitrary linear program  $\min\{c^T x \mid Ax \geq b, x \geq 0\}$  and a polytope  $\mathcal{U}$ , we call the system

$$\text{Minimize } \sum_{j=1}^n c_j x_j \quad (11.12)$$

$$\text{Subject to } \sum_{j=1}^n a_{ij} x_j \geq b_i \quad \forall i = 1, \dots, m, \quad \forall (a_{ij})_{i,j=1}^{m,n} \in \mathcal{U} \quad (11.13)$$

$$x_j \geq 0 \quad \forall j = 1, \dots, n \quad (11.14)$$

the robust counterpart of our original linear program. In order to model an uncertain right-hand side  $b$  of a linear program as well, we can introduce a fixed auxiliary variable and move  $b$  to the constraint matrix.

Polynomial time optimization over the feasible region of (11.12)–(11.14) is equivalent to having a polynomial time separation algorithm for the feasible region. This is an algorithm that decides whether a given point  $x \in \mathbf{R}_{\geq 0}^n$  is feasible for (11.12)–(11.14) and if not, yields a scenario  $A \in \mathcal{U}$  and an index  $i \in 1, \dots, m$  such that  $\text{row}_i(A)^T x < b_i$ , where  $\text{row}_i(A)$  denotes the  $i$ -th row of  $A$ . Such a separation algorithm for the robust counterpart exists if a separation oracle for  $\mathcal{U}$  exists; indeed, to separate  $x^*$  from the feasible region of (11.12)–(11.14), it suffices to solve

$$b_i^* := \min\{\text{row}_i(A)^T x^* \mid A \in \mathcal{U}\} \quad (11.15)$$

for all  $i = 1, \dots, m$ . If for some  $i$  we have  $b_i^* < b_i$ , then there is a separating inequality  $\text{row}_i(A)^T x \geq b_i$ . Yet, solving (11.15) is possible in polynomial time if and only if there is a polynomial time separation algorithm for  $\mathcal{U}$ .

**Theorem 1 (Ben-Tal and Nemirovski (1999))** *Let  $\mathcal{U} \subseteq \mathbf{R}^{m \times n}$  and let  $b \in \mathbf{R}^m$ . Then the robust counterpart*

$$\min\{c^T x \mid Ax \geq b, x \geq 0, \forall A \in \mathcal{U}\}$$

is solvable in time polynomial in  $m$  and  $n$  if there is separation algorithm for  $\mathcal{U}$  with a running time polynomial in  $m$  and  $n$ .  $\square$

Thus, we can solve worst-case robust linear programs polynomially if we can separate polynomially over the uncertainty set.

If the uncertainty polytope is given in its vertex description  $\mathcal{U} = \text{conv}\{A_1, \dots, A_k\}$  then separation is always possible in time polynomial in  $n$ ,  $m$  and  $k$  (although  $k$  may be exponentially large in  $m$  or  $n$ ). We say that  $\mathcal{U}$  is a *discrete set* in this case.

If the uncertainty polytope is described by a system of linear inequalities, a dualization approach similar to the budget uncertainty case can be applied.

## 2.6 Multi-stage Robustness

In the classical worst-case robust model, we take all the decisions in a *single stage* before we know the realization of the uncertain parameters. This modelling is not always desirable: We could instead imagine fixing some variables here-and-now while adjusting other variables once (part of) the uncertain parameters have realized. As an example in the network design context, we might decide on the capacities in a first stage (without knowing the realizations of the uncertain demands) and postpone the routing of the traffic to a point in time where the demands are known with certainty. Or, we might conservatively *buy* parts of the network in advance and *rent* additional capacity as needed. Depending on fewer uncertain parameters, such *multi-stage* models allow for less expensive solutions without sacrificing their robustness. The price for the additional flexibility is a computationally harder model, as even models with two stages of robustness tend to be NP-hard to solve. A middle ground is achieved by assuming a tractable dependence between the uncertain parameters and the adjustable, later stage variables. Models with *affine robustness* assume that the adjustable variables can be computed as affine functions from the uncertain data. *Recoverable robustness* more generally asks for a tractable algorithm that computes feasible values for all adjustable variables given the first stage decisions and the parameter realization.

## 3 Robust Network Designs

In the sequel, we will apply the robust optimization approach to network design. As opposed to the introductory Chap. 2, we suppose that  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  is an *undirected* graph with a node set  $\mathcal{N}$  and a set of potential edges  $\mathcal{E} \subseteq \binom{\mathcal{N}}{2}$ . To emphasize the difference, we use the notation  $\{i, j\}$  to distinguish an undirected edge between  $i, j \in \mathcal{N}$  from the directed arcs  $(i, j)$  and  $(j, i)$ . Then, a flow  $f$  on  $\mathcal{G}$  assigns a flow value  $f_{ij}$  and  $f_{ji}$  to both possible orientations of each edge  $\{i, j\}$  and a capacity

vector  $u \in \mathbf{R}^E$  admits a flow  $f$  if  $f_{ij} + f_{ji} \leq u_{ij}$ . Following the robust optimization paradigm, we consider a setting where the supplies / demands of the nodes in the graph are uncertain. The *robust network design problem* is the task to select a capacity  $u_{ij} \geq 0$  for each edge  $\{i, j\}$  such that *all* possible demand realizations can be satisfied while minimizing the total costs for installing the capacities.

Exactly *how* the demands are satisfied will vary throughout the chapter. We will first consider some single-commodity models and then generalize to multiple commodities. Given that we optimize over multiple scenarios, another modeling choice arises: Do we need to fix the routing of the demands before we know which scenario will realize or are we allowed to select a suitable routing once we know the scenario realization? In the former case, we need to compute a *routing template*, i.e.  $[0, 1]$ -valued flow. Given an arc  $(i, j)$ , we interpret the template flow  $f_{ij} \in [0, 1]$  as the percentage of the demand that is routed along  $(i, j)$ . In the terms of the previous section, this yields a single-stage optimization problem and we say that we model a *static routing* in this case. If in the latter case, we are allowed to choose the routing *after* a scenario has realized, we perform a two-stage optimization. This is known as a *dynamic routing*. In general, dynamic routings offer more flexibility and thus, cheaper solutions. They are, however, much harder to compute and do not fit all practical applications. Models in between static and dynamic routing exist; see for instance Poss and Raack (2013).

## 4 Single-Commodity Formulations

In Chap. 2, Sect. 2, we defined a deterministic (i.e., non-robust) single-commodity flow by saying that every node has a demand  $w_i$  of the single commodity. We generalize this notion by introducing an uncertainty set  $\mathcal{U} \subseteq \mathbf{R}^{\mathcal{N}}$  such that any scenario  $w \in \mathcal{U}$  defines a demand  $w_i$  for each node. As before, we say that  $i \in \mathcal{N}$  has a supply of the single commodity *in the scenario*  $w$  if  $w_i > 0$ , we say that  $i$  has a *demand* of the commodity in scenario  $w$  if  $w_i < 0$  and that  $i$  is a transshipment node in scenario  $w$  if  $w_i = 0$ . It is possible for nodes to have a supply in one scenario and a demand in another which means that our partitioning of the nodes into origin nodes  $\mathcal{N}_w^o$ , destination nodes  $\mathcal{N}_w^d$  and transshipment nodes  $\mathcal{N}_w^t$  now depends on the scenario  $w \in \mathcal{U}$ . As before, we assume that the supplies and demands are balanced in all scenarios in  $\mathcal{U}$  and observe that if  $\sum_{i \in \mathcal{N}} w_i \neq 0$  for some  $w \in \mathcal{U}$ , then no flow can satisfy the supplies and demands simultaneously and the problem instance is infeasible.

Given  $\mathcal{G}$ , an edge-cost vector  $c \in \mathbf{R}_{\geq 0}^{\mathcal{E}}$  and an uncertainty set  $\mathcal{U} \subseteq \mathbf{R}^{\mathcal{N}}$ , the *Single-Commodity Capacitated Robust Network Design Problem* (SSCCRND) is the task to find an integral minimum-cost capacity vector  $u \in \mathbf{Z}^{\mathcal{E}}$  that admits a feasible flow for each  $w \in \mathcal{U}$  while minimizing the capacity installation costs  $\sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij}$ . I.e., it is a two-stage robust optimization problem. The capacity



vector  $u$  has to be determined before the realization of the demand vector  $w$  is known, but the flow can be scenario-specific.

The SSCCRND problem is NP-hard, even if  $\mathcal{U}$  is a discrete uncertainty set with three scenarios that only use demands  $w_i \in \{-1, 0, 1\}$  and that agree on a common origin node. The deterministic variant of the problem (i.e.,  $|\mathcal{U}| = 1$ ) is polynomial time solvable as a minimum-cost flow problem, however, which is in contrast to the deterministic SCFND in Chap. 2, Sect. 2. It is currently unknown if the SSCCRND problem with two scenarios is NP-hard.

#### 4.1 A Flow-Based Formulation

The general problem admits a flow-based formulation of the SSCCRND problem that is similar to the flow-based formulation in Chap. 2, Sect. 2. It has an integer capacity variable  $u_{ij}$  for each edge  $\{i, j\} \in \mathcal{E}$  and two continuous arc-flow variables  $f_{ij}^w, f_{ji}^w$  for each edge  $\{i, j\} \in \mathcal{E}$  and each scenario  $w \in \mathcal{U}$  (modeling the flow on  $\{i, j\}$  in scenario  $w$ ).

$$\text{Minimize } \sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij} \quad (11.16)$$

$$\text{Subject to } \sum_{\{i,j\} \in \mathcal{E}} (f_{ij}^w - f_{ji}^w) = w_i \quad \forall i \in \mathcal{N}, \forall w \in \mathcal{U} \quad (11.17)$$

$$f_{ij}^w + f_{ji}^w \leq u_{ij} \quad \forall \{i, j\} \in \mathcal{E}, \forall w \in \mathcal{U} \quad (11.18)$$

$$f_{ij}^w, f_{ji}^w \geq 0 \quad \forall \{i, j\} \in \mathcal{E}, \forall w \in \mathcal{U} \quad (11.19)$$

$$u_{ij} \in \mathbf{Z}_{\geq 0}^E \quad \forall \{i, j\} \in \mathcal{E} \quad (11.20)$$

This formulation is known as the *flow*-based formulation and matches the definition of SSCCRND exactly; any feasible solution defines a feasible flow  $f^w$  for all scenarios  $w \in \mathcal{U}$  along with minimum integer capacities that support the flows. The constraint matrix of formulation (11.16)–(11.20) is not totally unimodular and thus the integrality requirement for the capacity variables is necessary. Given integer values for  $u$ , however, we can always find a feasible  $f$  that is integer as well (provided  $w$  is integer), even though integrality of the scenario flows is not required in the definition of the SSCCRND problem.

Formulation (11.16)–(11.20) potentially has an infinite number of variables. Assuming  $\mathcal{U}$  is a polytope, to make it a finite formulation, we can equivalently replace  $\mathcal{U}$  by the set of its vertices. Still, not only may the number of vertices of  $\mathcal{U}$  be large, if  $\mathcal{U}$  is given in a linear description, it is non-trivial to compute all vertices of  $\mathcal{U}$  efficiently.

## 4.2 A Cut-Set-Based Formulation

We obtain a formulation of finite size by projecting out the flow variables in (11.16)–(11.20). The result is a *cut-set*-based formulation that has an integer capacity variable  $u_{ij}$  for each edge  $\{i, j\}$ . Before we introduce the formulation itself, let us define a robust cut-set-based inequality for the SSCCRND problem as a generalization of the cut-set-based inequalities (2.16) in Chap. 2. Consider any cut  $\mathcal{S} \subseteq \mathcal{N}$  and some scenario  $w \in \mathcal{W}$ . Any feasible choice of capacities  $u$  must satisfy  $\sum_{\{i,j\} \in (\mathcal{S}, \bar{\mathcal{S}})} u_{ij} \geq W_{\mathcal{S}}(w)$ , where  $W_{\mathcal{S}}(w) := |\sum_{i \in \mathcal{S}} w_i|$  is defined analogously to Chap. 2, Sect. 2.1. This is true for all  $w \in \mathcal{W}$  and thus, it is necessary that  $u$  satisfies

$$\sum_{\{i,j\} \in (\mathcal{S}, \bar{\mathcal{S}})} u_{ij} \geq \max_{w \in \mathcal{W}} W_{\mathcal{S}}(w). \quad (11.21)$$

Inequality (11.21) is a *robust cut-set-based inequality* for the SSCCRND problem.

It turns out that a capacity vector  $u$  is feasible if it satisfies the robust cut-set-based inequality (11.21) for all  $\mathcal{S} \subseteq \mathcal{N}$ . This yields the following cut-set-based formulation for the SSCCRND problem.

$$\text{Minimize} \quad \sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij} \quad (11.22)$$

$$\text{Subject to} \quad \sum_{\{i,j\} \in (\mathcal{S}, \bar{\mathcal{S}})} u_{ij} \geq \max_{w \in \mathcal{W}} \left| \sum_{i \in \mathcal{S}} w_i \right| \quad \forall \mathcal{S} \subseteq \mathcal{N} \quad (11.23)$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.24)$$

The cut-set-based formulation is equivalent to the flow-based formulation (11.16)–(11.20) in the sense that the (fractional) solutions of the projection of (11.16)–(11.20) onto the  $u$ -space are exactly those defined by (11.23)–(11.24). The linear programming bounds obtained from the relaxations of the flow-based formulation and the cut-set-based formulation are hence the same. A cut-set-based inequality (11.23) for a set  $\mathcal{S}$  with  $\max_{w \in \mathcal{W}} \sum_{i \in \mathcal{S}} w_i > 0$  defines a facet of the feasible region of (11.23)–(11.24) if both  $\mathcal{S}$  and the complement set  $\bar{\mathcal{S}}$  induce a connected subgraph of  $\mathcal{G}$ . The non-negativity constraint (11.24) for  $u_{ij}$ ,  $\{i, j\} \in \mathcal{E}$  is dominated by a cut-set-based inequality if removing  $\{i, j\}$  disconnects  $\mathcal{G}$  into two partitions with non-zero total balance. In all other cases, the constraint defines a facet of the feasible region of the cut-set-based formulation.

### 4.3 Separating Robust Cut-Set-Based Inequalities

The size of the cut-set-based formulation is independent of  $\mathcal{U}$ . However, the number of constraints in the formulation is exponential in the size of  $\mathcal{G}$ . Whether these constraints can be separated with sufficient efficiency depends on the uncertainty set; for general uncertainty sets, the separation is NP-hard. In this case, applying the generic transformation of the robust counterpart from Sect. 2.5 leads to a linear program with exponentially many constraints and variables. Neither a separation nor a pricing algorithm is known for this program. Alternatively, the problem can be reformulated as a non-convex quadratic program.

There are known tractable special cases, however. The first tractable case occurs when the vertices of  $\mathcal{U}$  can be enumerated efficiently (for instance, because  $\mathcal{U}$  is a discrete uncertainty set). Then, the separation requires one run of a minimum  $s$ - $t$ -cut algorithm per vertex of  $\mathcal{U}$ . We discuss two other tractable special case in more detail below.

#### 4.3.1 The Single-Commodity Hose Uncertainty Set

Hose uncertainty corresponds to the interval robustness model in the previous section, with the additional constraint that all scenarios must induce balanced supplies and demands. For each node  $i \in \mathcal{N}$ , we define a lower bound  $w_i^{\min} \in \mathbf{Z}$  and an upper bound  $w_i^{\max} \in \mathbf{Z}$  on the demand at  $i$  and consider any fluctuation of the demands that lies within these bounds and defines a balanced scenario. This results in the following uncertainty set.

$$\mathcal{U}_H(w^{\min}, w^{\max}) := \left\{ w \in \mathbf{R}^V \mid w_i \in [w_i^{\min}, w_i^{\max}] \forall i \in \mathcal{N} \wedge \sum_{i \in \mathcal{N}} w_i = 0 \right\}. \quad (11.25)$$

The SSCCRND problem with Hose uncertainties remains NP-hard, as does the separation problem for the robust cut-set-based inequalities. However, the separation problem can be reformulated as a mixed integer linear program (MILP) in the following way. The MILP computes a cut  $\mathcal{S}$  and the value of the right-hand side of the corresponding cut-set-based inequality. For each node  $i \in \mathcal{N}$ , we introduce a variable  $\pi_i$  indicating if  $i \in \mathcal{S}$  and a binary decision variable  $\rho_{ij}$  indicating if  $\{i, j\} \in (\mathcal{S}, \bar{\mathcal{S}})$ . An additional continuous variable  $W$  holds the right-hand side value of the cut-set-based inequality corresponding to  $\mathcal{S}$ . The MIP minimizes the slack of the cut-set-based inequality induced by  $\mathcal{S} = \{i \in \mathcal{N} \mid \pi_i = 1\}$ .

$$\text{Minimize} \quad \sum_{\{i,j\} \in E} u_{ij}^* \rho_{ij} - W \quad (11.26)$$

$$\text{Subject to} \quad W \leq \sum_{i \in \mathcal{N}} \pi_i w_i^{\max} \quad (11.27)$$

$$W \leq - \sum_{i \in \mathcal{N}} (1 - \pi_i) w_i^{\min} \quad (11.28)$$

$$\pi_i - \pi_j \leq \rho_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.29)$$

$$\pi_j - \pi_i \leq \rho_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.30)$$

$$\pi_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \quad (11.31)$$

$$\rho_{ij} \in \{0, 1\} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.32)$$

Here, the essential insight is that we can assume without loss of generality that the minimum slack is attained by a cut-set-based inequality for a set  $\mathcal{S}$  where  $\max_{w \in \mathcal{U}_H} \sum_{i \in \mathcal{S}} w_i$  is non-negative (otherwise, replace  $\mathcal{S}$  by the complement  $\bar{\mathcal{S}}$ ). This observation implies that the right-hand side of the inequality simplifies to

$$\max_{w \in \mathcal{U}_H} \left| \sum_{i \in \mathcal{S}} w_i \right| = \max \left\{ \sum_{i \in \mathcal{S}} w_i^{\max}, - \sum_{i \in \bar{\mathcal{S}}} w_i^{\min} \right\}, \quad (11.33)$$

which is modeled by the constraints (11.27) and (11.28). Computational experiments show that this MILP is solvable for reasonable instance sizes.

### 4.3.2 Network Containment

The *network containment* uncertainty polytope defines another tractable special case of the SSCCRND problem. In this model, the demand is not defined directly at each node, but through *demand requests*: For each pair of nodes  $i, j \in \mathcal{N}$ , we define a minimum and a maximum amount  $r_{ij}^{\min}, r_{ij}^{\max}$ , respectively, of the global single commodity that  $j$  can request from  $i$ . We then project the demand requests down onto node demands by writing the corresponding uncertainty polytope as

$$\mathcal{U}_C := \left\{ \left( \sum_{j \in \mathcal{N}} (r_{ij} - r_{ji}) \right)_{i \in \mathcal{N}} \in \mathbf{R}^{\mathcal{N}} \mid r_{ij}^{\min} \leq r_{ij} \leq r_{ij}^{\max} \forall i, j \in \mathcal{N} \right\} \quad (11.34)$$

Thus, for any choice of demand requests  $r$ , we obtain a scenario  $w$  where the demand  $w_i$  at node  $i$  is exactly the total amount  $\sum_{j \in \mathcal{N}} r_{ij}$  of the commodity requested from  $i$  minus the total amount  $\sum_{j \in \mathcal{N}} r_{ji}$  of the commodity requested by  $i$ . In contrast to the Hose model, the scenarios defined in this way are always balanced even without an explicit balancing constraint. The definition also implies that at any node  $i$ , the demand request  $r_{ij}$  may be satisfied by any node  $j'$  (i.e., in general, we may have  $j' \neq j$ ) as long as  $i$  receives or sends the correct amount of the commodity. The cut-set separation problem for the network containment uncertainty polytope can be solved by a mixed integer linear program similarly to the Hose polytope.

#### 4.4 Strengthening the Formulations

The feasible regions of the flow-based and of the cut-set-based formulation admit a project-and-lift cut generation procedure that works by contracting edges in an SSCCRND instance. To contract any edge  $\{i, j\} \in E$ , we merge  $i$  and  $j$  into a super node  $i'$ . All nodes that were previously adjacent to  $i$  or  $j$  are now adjacent to  $i'$  and we delete all resulting parallel edges. In all scenarios  $w \in \mathcal{U}$ , we set the demand of  $i'$  to be  $w_i + w_j$ . By applying this procedure repeatedly, we can project any SSCCRND instance  $I$  into an instance  $I'$  with fewer nodes and edges. This smaller instance has nice properties: Any valid inequality for  $I'$  can be turned into a valid inequality for  $I$  by appropriately lifting the coefficients and moreover, the lifting procedure ensures that facet-defining inequalities for  $I'$  remain facet-defining for  $I$ .

To find valid inequalities for  $I'$ , we can repeatedly contract edges until we obtain an instance of constant size. We then apply the *target cut approach* that finds facet-defining inequalities by solving a linear program that has an inequality for each vertex of the feasible region of  $I'$ . In the special case where  $I'$  is a triangle graph, its three super-nodes define a partitioning of the node set into three disjoint sets  $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 = \mathcal{N}$ . This partitioning gives rise to the class of 3-partition inequalities:

$$\sum_{\{i,j\} \in (\mathcal{S}_1:\mathcal{S}_2)} u_{ij} + \sum_{\{i,j\} \in (\mathcal{S}_1:\mathcal{S}_3)} u_{ij} + \sum_{\{i,j\} \in (\mathcal{S}_2:\mathcal{S}_3)} u_{ij} \geq \left\lceil \frac{W_{\mathcal{S}_1} + W_{\mathcal{S}_2} + W_{\mathcal{S}_3}}{2} \right\rceil. \quad (11.35)$$

Here, for any  $\mathcal{S}, \mathcal{S}' \in \mathcal{N}$ , we define  $(\mathcal{S} : \mathcal{S}') := \{\{i, j\} \in \mathcal{E} \mid i \in \mathcal{S} \wedge j \in \mathcal{S}'\}$  to be the set of edges with one endpoint in  $\mathcal{S}$  and one endpoint in  $\mathcal{S}'$ . Further, we let  $W_{\mathcal{S}} := \max_{w \in \mathcal{U}} W_{\mathcal{S}}(w)$  be the right-hand side value of the cut-set-based inequality induced by  $\mathcal{S}$ . The 3-partition inequalities are facet-defining for the feasible region defined by the cut-set-based formulation (11.22)–(11.24) if  $W_{\mathcal{S}_1} + W_{\mathcal{S}_2} + W_{\mathcal{S}_3} > 0$  is odd and each of the sets  $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$  induces a connected subgraph. The 3-partition inequality corresponding to  $\mathcal{S}_1, \mathcal{S}_2$  and  $\mathcal{S}_3$  can be generated as a  $\{0, \frac{1}{2}\}$ -Chvátal-Gomory cuts from the cut-set-based inequalities corresponding to  $\mathcal{S}_1, \mathcal{S}_2$  and  $\mathcal{S}_1 \cup \mathcal{S}_2$ .

#### 4.5 Variants of the Problem

It is straight-forward to rewrite (11.22)–(11.24) for a setting where opening an edge  $\{i, j\}$  incurs a *fixed-cost* of  $c_{ij}$ , but provides a fixed capacity  $u_{ij}$ . The variant with fixed costs and *uncapacitated edges* can then be seen as the special case where the fixed capacities are set to the sum  $W = \sum_{i \in \mathcal{N}} \max_{w \in \mathcal{U}} |w_i|$  of the maximum demands. If  $\mathcal{U}$  is described by a system of inequalities, this value can be obtained by solving a linear program for each node  $i \in \mathcal{N}$ . Transportation costs (in the worst-case) can be added to the objective function of the flow-based formulation (11.16)–

(11.20). Neither variants with fixed costs, transportation costs, nor uncapacitated variants have been considered in the literature so far, to the best of our knowledge.

## 5 Multicommodity Formulations

If the traffic between different pairs of nodes in our network has to be distinguished, then the multicommodity flow model is an appropriate modeling choice. Analogously to Chap. 2, Sect. 3 we assume that each commodity  $k \in \mathcal{K}$  is given as an origin-destination pair  $(O(k), D(k)) \in \mathcal{N} \times \mathcal{N}$ . To model the uncertain demands, we consider an uncertainty polytope  $\mathcal{U} \subseteq \mathbf{R}^{\mathcal{K}}$ . Any scenario  $d \in \mathcal{U}$  specifies a demand  $d^k$  for commodity  $k \in \mathcal{K}$ . As before, we can re-write the demands as flow balances by setting

$$w_i^{k,d} := \begin{cases} d^k, & \text{if } i = O(k) \\ -d^k, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall d \in \mathcal{U}, \forall k \in \mathcal{K}, \forall i \in \mathcal{N}. \quad (11.36)$$

Here, however, the uncertainty polytope introduces a dependence of  $w$  on the scenario  $d$ . Given an undirected graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  with commodities  $(O(k), D(k))$  for  $k \in \mathcal{K}$ , a scenario polytope  $\mathcal{U} \subseteq \mathbf{R}^{\mathcal{K}}$ , and an installation cost  $c_{ij}$  for each edge  $e \in \mathcal{E}$ , the MSCCRND problem is to find an integer capacity  $u_{ij}$  for each edge  $\{i, j\}$  such that  $\sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij}$  is minimum and all demands in  $\mathcal{U}$  can be routed. In the dynamic routing case, the demands must be routed with a multicommodity flow. In the case of static routing, the demands are routed with a routing template as briefly described in Sect. 3.

### 5.1 Standard Uncertainty Sets

Consider the application of the budget uncertainty approach to multicommodity network design (cf. Sect. 2.4). We denote by  $\bar{d}^k$  a nominal value for the demand of each commodity  $k \in \mathcal{K}$  and we suppose that the true demand of the commodity can deviate from its nominal value by at most  $\hat{d}^k$ . Additionally, there can be at most  $\Gamma \in \mathbf{Z}_{\geq 0}$  deviations at the same time. We define the resulting uncertainty set as the  $\Gamma$ -robustness polytope for the MSCCRND.

$$\mathcal{U}_B(\bar{d}, \hat{d}, \Gamma) := \text{conv} \left\{ d \in \mathbf{R}_{\geq 0}^K \left| \begin{array}{ll} d^k \in [\bar{d}^k - \hat{d}^k, \bar{d}^k + \hat{d}^k], & \text{if } k \in S \\ d^k = \bar{d}^k, & \text{otherwise} \end{array} \right. \right. \\ \left. \left. \forall S \subseteq \mathcal{K} \text{ with } |S| \leq \Gamma \right. \right\}. \quad (11.37)$$

Defining  $S(\Gamma) := \{\sigma \in [0, 1]^K \mid \sum_{k \in \mathcal{K}} \sigma^k \leq \Gamma\}$  as the set of possible deviations, we can rewrite the  $\Gamma$ -robustness polytope equivalently as the following set.

$$\mathcal{U}_B(\bar{d}, \hat{d}, \Gamma) = \bar{d} + \{(\sigma^k \hat{d}^k)_{k \in \mathcal{K}} \in \mathbf{R}_{\geq 0}^K \mid \sigma \in S(\Gamma)\}. \quad (11.38)$$

An alternative to the budget uncertainty set was proposed by Fingerhut et al. (1997) and Duffield et al. (1999) independently. In the *Hose* model, we only assume that we know the maximum *incoming* traffic  $d_i^{\text{in}}$  and the maximum *outgoing* traffic  $d_i^{\text{out}}$  at each node  $i \in V$ . We then define a commodity  $(s, t)$  with demand  $d^{st}$  for all pairs of nodes  $s, t \in \mathcal{N}$  and consider any demand vector  $d$  that adheres to the traffic bounds. We call the resulting uncertainty set

$$\mathcal{U}_H(d^{\text{in}}, d^{\text{out}}) := \left\{ d \in \mathbf{R}_{\geq 0}^{\mathcal{N} \times \mathcal{N}} \mid \sum_{t \in \mathcal{N}} d^{it} \leq d_i^{\text{out}} \wedge \sum_{s \in \mathcal{N}} d^{si} \leq d_i^{\text{in}} \forall i \in \mathcal{N} \right\} \quad (11.39)$$

the (multicommodity) *Hose* polytope. We speak of the symmetric *Hose* polytope if  $d_i^{\text{in}} = d_i^{\text{out}}$  for all nodes  $i \in \mathcal{N}$ .

For this uncertainty model, we only need to estimate  $\Theta(|\mathcal{N}|)$  parameters (as opposed to a worst case of  $\Theta(|\mathcal{N}|^2)$  in the budget uncertainty case). Additionally, these parameters are easier to predict and can even be known exactly if they stem from technical specifications or legal contracts.

## 5.2 The VPN Problem

The MSCCRND problem with *Hose* uncertainties and static routing is known as the *Virtual Private Network* (VPN) design problem. The problem is NP-hard in general, but can be solved efficiently if the *Hose* polytope is symmetric in the above sense and if we force the flow to be unsplittable. In that case, there is always an optimum routing template that forms a tree and optimum *tree routing templates* for unsplittable flows can be found in polynomial time. We will see in the following section how the general VPN problem can be solved with ILP methods.

### 5.3 Static Routing: Arc-Flow Based Formulations

In the static routing case, the flow formulation does not need a set of arc-flow variables for every scenario: As we use the same routing template in all scenarios, a single set is sufficient. Here, for all commodities  $k \in \mathcal{K}$  and all edges  $\{i, j\} \in \mathcal{E}$ , the routing template variables  $f_{ij}^k$  and  $f_{ji}^k$  denote the fraction of the demand of the commodity  $k$  that is routed via the arcs  $(i, j)$  and  $(j, i)$ , respectively, in each scenario  $d \in \mathcal{U}$ . As before, we use an integer variable  $u_{ij}$  to model the capacity of the edge  $\{i, j\}$ , for all  $\{i, j\} \in \mathcal{E}$ .

$$\text{Minimize } \sum_{\{i,j\} \in E} c_{ij} u_{ij} \quad (11.40)$$

$$\text{Subject to } \sum_{\{i,j\} \in E} f_{ij}^k - f_{ji}^k = \begin{cases} 1, & \text{if } i = O(k) \\ -1, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} \forall k \in \mathcal{K} \\ \forall i \in \mathcal{N} \end{matrix} \quad (11.41)$$

$$\sum_{k \in \mathcal{K}} d^k \cdot (f_{ij}^k + f_{ji}^k) \leq u_{ij} \quad \begin{matrix} \forall \{i, j\} \in \mathcal{E} \\ \forall d \in \mathcal{U} \end{matrix} \quad (11.42)$$

$$f_{ij}^k, f_{ji}^k \in [0, 1] \quad \begin{matrix} \forall \{i, j\} \in \mathcal{E} \\ \forall k \in \mathcal{K} \end{matrix} \quad (11.43)$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.44)$$

Again, it is sufficient to include the constraints (11.42) for the vertices of  $\mathcal{U}$ . Forcing that  $f$  is binary leads to a variant where the template for each commodity uses a unique path and thus, a routing template for an unsplittable flow.

Let us now robustify the arc-flow formulation (11.40)–(11.44) with the  $\Gamma$ -robustness model.

$$\text{Minimize } \sum_{\{i,j\} \in E} c_{ij} u_{ij} \quad (11.45)$$

Subject to

$$\sum_{\{i,j\} \in \mathcal{E}} f_{ij}^k - f_{ji}^k = \begin{cases} 1, & \text{if } i = O(k) \\ -1, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} \forall k \in \mathcal{K} \\ \forall i \in V \end{matrix} \quad (11.46)$$

$$\sum_{k \in \mathcal{K}} \bar{d}^k (f_{ij}^k + f_{ji}^k) + \max_{\sigma \in \mathcal{U}_B(\Gamma)} \sum_{k \in \mathcal{K}} \sigma_k \hat{d}^k (f_{ij}^k + f_{ji}^k) \leq u_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.47)$$



$$f_{ij}^k, f_{ji}^k \in [0, 1] \quad \forall \{i, j\} \in \mathcal{E} \quad (11.48)$$

$$\forall k \in \mathcal{K}$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.49)$$

The constraint (11.47) contains an optimization problem in itself so that the formulation is a two level program. We would like to collapse the program into a single level. Unfortunately, in the inner level inside of the constraint (11.47), we seek to *maximize*  $\sum_{k \in \mathcal{K}} \sigma_k \hat{d}^k(f_{ij}^k + f_{ji}^k)$  over  $\mathcal{U}_B(\Gamma)$  while the outer level (11.45)–(11.48) strives to minimize this value. If we could rewrite the maximization as an equivalent minimization problem, we could collapse the two levels as desired. Given fixed template flows  $f^k$ , this can be achieved by modeling  $\max_{\sigma \in \mathcal{U}_B(\Gamma)} \sum_{k \in \mathcal{K}} \sigma_k \hat{d}^k(f_{ij}^k + f_{ji}^k)$  as a linear program and replacing it by its dual

$$\text{Minimize } \gamma_{ij} \cdot \Gamma + \sum_{k \in \mathcal{K}} \tau_{ij}^k \quad (11.50)$$

$$\text{Subject to } \gamma_{ij} + \tau_{ij}^k \geq \hat{d}^k(f_{ij}^k + f_{ji}^k) \quad \forall k \in \mathcal{K} \quad (11.51)$$

$$\tau_{ij}^k \geq 0 \quad \forall k \in \mathcal{K} \quad (11.52)$$

$$\gamma_{ij} \geq 0 \quad (11.53)$$

for all edges  $\{i, j\} \in \mathcal{E}$ . The result is a compact mixed-integer linear program with  $\Theta(|\mathcal{N}|^2|\mathcal{E}|)$  variables and  $\Theta(|\mathcal{N}|^3 + |\mathcal{N}|^2|\mathcal{E}|)$  constraints.

$$\text{Minimize } \sum_{\{i, j\} \in \mathcal{E}} c_{ij} u_{ij} \quad (11.54)$$

$$\text{Subject to } \sum_{\{i, j\} \in \mathcal{E}} f_{ij}^k - f_{ji}^k = \begin{cases} 1, & \text{if } i = O(k) \\ -1, & \text{if } i = D(k) \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} \forall k \in \mathcal{K} \\ \forall i \in \mathcal{N} \end{matrix} \quad (11.55)$$

$$\sum_{k \in \mathcal{K}} \bar{d}^k(f_{ij}^k + f_{ji}^k) + \gamma_{ij} \cdot \Gamma + \sum_{k \in \mathcal{K}} \tau_{ij}^k \leq u_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.56)$$

$$\gamma_{ij} + \tau_{ij}^k - \hat{d}^k(f_{ij}^k + f_{ji}^k) \geq 0 \quad \begin{matrix} \forall k \in \mathcal{K} \\ \forall \{i, j\} \in \mathcal{E} \end{matrix} \quad (11.57)$$

$$\tau_{ij}^k \geq 0 \quad \begin{matrix} \forall k \in \mathcal{K} \\ \forall \{i, j\} \in \mathcal{E} \end{matrix} \quad (11.58)$$

$$\gamma_{ij} \geq 0 \quad \forall \{i, j\} \in \mathcal{E} \quad (11.59)$$

$$f_{ij}^k, f_{ji}^k \in [0, 1] \quad \forall k \in \mathcal{K} \quad (11.60)$$

$$\forall \{i, j\} \in \mathcal{E}$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.61)$$

As the objective function makes sure that the left-hand side of constraint (11.56) is minimized, we can omit the explicit minimization without inserting complementary slackness conditions.

Alternatively, we can solve the separation problem for the constraints (11.47). Given fixed  $f$  and  $u$ , the problem amounts to finding a deviation  $\sigma \in \mathcal{U}_B(\Gamma)$  and an edge  $\{i, j\} \in E$  such that

$$\sum_{k \in \mathcal{K}} \bar{d}^k(f_{ij}^k + f_{ji}^k) + \sum_{k \in \mathcal{K}} \sigma^k \hat{d}^k(f_{ij}^k + f_{ji}^k) > u_{ij} \quad (11.62)$$

or to decide that none such combination of a deviation and an edge exists. The problem can be solved separately for each fixed edge  $\{i, j\} \in \mathcal{E}$ . Then, it amounts to solving

$$\max_{\sigma \in \mathcal{U}_B(\Gamma)} \sum_{k \in \mathcal{K}} \sigma^k \hat{d}^k(f_{ij}^k + f_{ji}^k). \quad (11.63)$$

If the optimum value of (11.63) is larger than  $u_{ij} - \sum_{k \in \mathcal{K}} \bar{d}^k(f_{ij}^k + f_{ji}^k)$ , then we found a violated inequality; otherwise, no violated inequality involving the edge  $\{i, j\}$  exists. To solve (11.63), we can sort the values  $\hat{d}^k(f_{ij}^k + f_{ji}^k)$  for  $k \in \mathcal{K}$  in non-increasing order. Then, the first  $\Gamma$  commodities determine a worst-case deviation. This approach yields a program that initially has  $\Theta(|\mathcal{K}| \cdot |\mathcal{N}|)$  constraints and  $\Theta(|\mathcal{K}| \cdot |\mathcal{E}|)$  variables. To solve the linear programming relaxation of the problem, we need to solve  $\Theta(|\mathcal{E}|)$  separation problems per iteration of the separation algorithm.

To combine the arc-flow formulation with the Hose polytope, we can equivalently replace constraint (11.42) by the optimization

$$u_{ij} = \max_{s, t \in V} \sum d^{st}(f_{ij}^{st} + f_{ji}^{st}) \quad (11.64)$$

$$\text{Subject to} \quad \sum_{t \in V} d^{st} \leq d_s^{\text{out}} \quad \forall s \in V \quad (11.65)$$

$$\sum_{s \in V} d^{st} \leq d_t^{\text{in}} \quad \forall t \in V \quad (11.66)$$

$$d^{st} \geq 0 \quad \forall s, t \in V \quad (11.67)$$

for all  $\{i, j\} \in \mathcal{E}$ . For fixed  $f$ , this gives us a bounded, feasible linear program for each edge  $\{i, j\} \in E$ . Again, we now replace these programs by their dual. In the linear program for edge  $\{i, j\}$ , denote by  $\omega_s^{ij}$  and  $v_t^{ij}$  the dual variables

corresponding to the constraints (11.65) and (11.66), respectively. This yields the following dual for each edge  $\{i, j\} \in \mathcal{E}$ .

$$\text{Minimize } \sum_{s \in \mathcal{N}} d_s^{\text{out}} \omega_s^{ij} + \sum_{t \in \mathcal{N}} d_t^{\text{in}} v_t^{ij} \quad (11.68)$$

$$\text{Subject to } \omega_s^{ij} + v_t^{ij} \geq f_{ij}^{st} + f_{ji}^{st} \quad \forall s, t \in V \quad (11.69)$$

$$\omega_s^{ij} \geq 0 \quad \forall s \in V \quad (11.70)$$

$$v_t^{ij} \geq 0 \quad \forall t \in V \quad (11.71)$$

This program is linear, even for a non-fixed  $f$ . We insert it into formulation (11.40)–(11.44), replacing constraint (11.41).

$$\text{Minimize } \sum_{\{i,j\} \in E} c_{ij} u_{ij} \quad (11.72)$$

$$\text{Subject to } \sum_{\{i,j\} \in E} f_{ij}^{st} - f_{ji}^{st} = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = t \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} \forall s, t \in \mathcal{N} \\ \forall i \in \mathcal{N} \end{matrix} \quad (11.73)$$

$$\sum_{s \in \mathcal{N}} d_s^{\text{out}} \omega_s^{ij} + \sum_{t \in \mathcal{N}} d_t^{\text{in}} v_t^{ij} \leq u_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.74)$$

$$\omega_s^{ij} + v_t^{ij} \geq f_{ij}^{st} + f_{ji}^{st} \quad \begin{matrix} \forall \{i, j\} \in \mathcal{E} \\ \forall s, t \in \mathcal{N} \end{matrix} \quad (11.75)$$

$$\omega_s^{ij}, v_s^{ij} \geq 0 \quad \begin{matrix} \forall \{i, j\} \in \mathcal{E} \\ \forall s \in \mathcal{N} \end{matrix} \quad (11.76)$$

$$f_{ij}^{st}, f_{ji}^{st} \in [0, 1] \quad \begin{matrix} \forall \{i, j\} \in \mathcal{E} \\ \forall s, t \in \mathcal{N} \end{matrix} \quad (11.77)$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.78)$$

In this way, we directly obtain a single level mixed integer linear program and do not need any further linearization. Solving the program gives us minimum cost integer capacities for MSCCRND with static routing over the Hose polytope. The program has  $\Theta(|\mathcal{N}|^2|\mathcal{E}|)$  variables and  $\Theta(|\mathcal{N}|^3 + |\mathcal{N}|^2|\mathcal{E}|)$  constraints. A similar approach for a problem variant with multiple facilities was given by Altin et al. (2011). Requiring that  $f$  is integral yields an MIP formulation for the VPN problem.

### 5.4 Static Routing: Path Based Formulations

In the static routing case, the MSCCRND problem with an arbitrary uncertainty set  $\mathcal{U}$  can be cast into a path-formulation. In this formulation, we additionally assume that there is an upper bound  $\bar{u}_{ij}$  on the capacity  $u_{ij}$  of edge  $\{i, j\} \in \mathcal{E}$  so that the feasible region is bounded. If these bounds are not desired, they can be replaced by a sufficiently large number (for instance, we can set the upper bound to  $\sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} \max_{d \in \mathcal{U}} d_i^k$  for all edges). For ease of notation, we let  $\mathcal{P}_k$  be the set of all paths between the origin  $O(k)$  of the  $k$ -th commodity and its destination  $D(k)$ , for all  $k \in \mathcal{K}$ . As before, let  $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$ . We use a continuous path variable  $x_p$  for each  $p \in \mathcal{P}$ .

$$\text{Minimize } \sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij} \quad (11.79)$$

$$\text{Subject to } \sum_{p \in \mathcal{P}_k} x_p = 1 \quad \forall k \in \mathcal{K} \quad (11.80)$$

$$\sum_{k \in \mathcal{K}} \sum_{\substack{p \in \mathcal{P}_k: \\ \{i,j\} \in p}} d^k x_p \leq u_{ij} \quad \begin{array}{l} \forall \{i, j\} \in \mathcal{E} \\ \forall d \in \mathcal{U} \end{array} \quad (11.81)$$

$$x_p \in [0, 1] \quad \forall p \in \mathcal{P} \quad (11.82)$$

$$u_{ij} \in \{0, \dots, \bar{u}_{ij}\} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.83)$$

To model unsplittable routing it suffices to turn the path-flow variables  $x$  in program (11.79)–(11.83) into binary variables.

We now decompose the path formulation (11.79)–(11.83) into a master and two satellite problems. The *master* problem consists of a variant of the path formulation. It maintains a set  $\bar{\mathcal{P}} \subseteq \mathcal{P}$  of relevant paths as well as a set of relevant scenarios  $\bar{\mathcal{U}} \subseteq \mathcal{U}$ . Both sets are initially empty.

$$\text{Minimize } \sum_{\{i,j\} \in E} c_{ij} u_{ij} \quad (11.84)$$

$$\text{Subject to } \sum_{p \in \bar{\mathcal{P}}_k} x_p \geq 1 \quad \forall k \in \mathcal{K} \quad (11.85)$$

$$\sum_{\substack{p \in \bar{\mathcal{P}}_k: \\ \{i,j\} \in p}} x_p \leq f_{ij}^k \quad \forall k \in \mathcal{K}, \quad \forall \{i, j\} \in \mathcal{E} \quad (11.86)$$

$$\sum_{k \in \mathcal{K}} f_{ij}^k d^k \leq u_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad \forall (d_{st})_{s,t \in V} \in \bar{\mathcal{U}} \quad (11.87)$$

$$u_{ij} \leq \bar{u}_{ij} \quad \forall \{i, j\} \in \mathcal{E} \quad (11.88)$$

$$x_p \in [0, 1] \quad \forall p \in \bar{\mathcal{P}} \quad (11.89)$$

$$f_{ij}^k \in [0, 1] \quad \forall k \in \mathcal{K}, \quad \forall \{i, j\} \in \mathcal{E} \quad (11.90)$$

The master problem (11.84)–(11.90) is bounded. Suppose for the moment that it is feasible as well and let  $u^*$  be an optimum solution for the problem. In order to guarantee that  $u^*$  is globally optimum, we need to make sure that adding additional paths to  $\bar{\mathcal{P}}$  cannot improve the value of  $u^*$ . Moreover,  $u^*$  must be globally feasible, i.e., the capacities must be sufficient to route all scenarios in  $\mathcal{U}$  (and not only those in  $\bar{\mathcal{U}}$ ). For the former problem, we solve a *path satellite problem*.

It consists of computing a shortest path between all origin-destination pairs with respect to the dual variables  $\pi$  and  $\rho$  of the constraints (11.85) and (11.86). Indeed, one can argue that if  $\sum_{\{i,j\} \in p} \rho_{ij}^k < \pi^k$  for some  $O(k)$ - $D(k)$ -path  $p \notin \bar{\mathcal{P}}$ , then  $p$  will improve the current solution  $u^*$ . In this case, we add the path  $p$  to  $\bar{\mathcal{P}}$ . To ensure global feasibility on the other hand, we separate inequalities of type (11.87) in a *demand satellite problem*. Given fixed routing variables  $f$  and fixed capacities  $u$  from an optimum solution of (11.84)–(11.90), it suffices to solve the linear program

$$u_{ij}^{\max} := \max \sum_{k \in \mathcal{K}} f_{ij}^k d^k \quad (11.91)$$

$$\text{Subject to} \quad d \in \mathcal{U} \quad (11.92)$$

for all edges  $\{i, j\} \in E$ . Notice that here, we optimize over the entire scenario set. If for some edge  $\{i, j\} \in E$  we find that  $u_{ij}^{\max} > u_{ij}$ , then the inequality

$$\sum_{k \in \mathcal{K}} f_{ij}^k d^k \leq u_{ij} \quad (11.93)$$

is violated by  $(f, u)$ . We add the corresponding optimum solution of (11.91)–(11.92) to  $\bar{\mathcal{U}}$ , thus adding the violated inequality (11.93) to the master problem.

To solve the master problem to global optimality, it now suffices to iteratively call the path satellite, the demand satellite and the master problem itself until neither new paths nor new scenarios are found. If at some point during the computation the master problem becomes infeasible, we call the path satellite and if no improving paths can be found, then the problem instance must be globally infeasible (i.e., the upper bounds for the capacities are too restrictive to route all scenarios in  $\mathcal{U}$ ).

## 5.5 Dynamic Routing: Arc-Flow Based Formulations

The robustification of the capacitated multicommodity network design problem works analogously to the SSCCRND case. For the arc-flow formulation, we

introduce one set of arc-flow variables for each scenario  $d \in \mathcal{U}$  and each commodity  $k \in \{1, \dots, K\}$ . This gives us a robustified version of the classical arc-flow formulation.

$$\text{Minimize } \sum_{\{i,j\} \in E} c_{ij} u_{ij} \tag{11.94}$$

$$\text{Subject to } \sum_{\{i,j\} \in E} f_{ij}^{k,d} - f_{ji}^{k,d} = w_i^{k,d}, \quad \forall i \in V, \forall d \in \mathcal{U}, \quad \forall k \in \mathcal{K} \tag{11.95}$$

$$\sum_{k \in \mathcal{K}} f_{ij}^{k,d} + f_{ji}^{k,d} \leq u_{ij}, \quad \forall \{i, j\} \in \mathcal{E}, \quad \forall d \in \mathcal{U}, \tag{11.96}$$

$$f_{ij}^{k,d}, f_{ji}^{k,d} \geq 0, \quad \forall k \in \mathcal{K}, \quad \forall d \in \mathcal{U}, \forall \{i, j\} \in \mathcal{E} \tag{11.97}$$

$$u_{ij} \in \mathbf{Z}_{\geq 0}^E, \quad \forall \{i, j\} \in \mathcal{E} \tag{11.98}$$

Featuring  $2|\mathcal{K}||\mathcal{E}|$  flow-variables for each scenario  $d \in \mathcal{U}$  and  $\Theta(|\mathcal{E}|)$  constraints (11.96) for all  $d \in \mathcal{U}$ , this formulation is of infinite size. Again, if  $\mathcal{U}$  is a polytope, we can replace  $\mathcal{U}$  by the set of its vertices to obtain a finite (although potentially inpractical) formulation.

### 5.6 Dynamic Routing: Formulations Without Flow Variables

As also can be observed for deterministic multi-commodity network design problems, Gale’s cut condition is not sufficient for the existence of a multi-commodity flow and thus, these problems cannot be cast into a cut-set based formulation in general. There is, however, a generalization of Gale’s condition, called the *Japanese Theorem*, by Onaga and Kakusho (1971) that enables us to formulate the problem with capacity variables only. Let  $\mathcal{M} \subseteq \mathbf{R}_{\geq 0}^{\mathcal{N} \times \mathcal{N}}$  be the metric cone, i.e., set of all real metrics on  $\mathcal{N}$ . Given capacities  $u \in \mathbf{R}_{\geq 0}^{\mathcal{E}}$ , a feasible multicommodity flow exists if and only if

$$\sum_{\{i,j\} \in \mathcal{E}} \mu_{ij} u_{ij} \geq \sum_{k \in \mathcal{K}} d^k \cdot \text{dist}_{\mu}(O(k), D(k)) \quad \forall \text{metrics } \mu \in \mathcal{M}, \tag{11.99}$$

where  $\text{dist}_{\mu}(s, t)$  denotes the shortest path distance from  $s \in \mathcal{N}$  to  $t \in \mathcal{N}$  with respect to  $\mu$ . To see why the condition is necessary, observe the following: Let  $\mu \in \mathcal{M}$  be any metric and let us interpret  $\mu$  as edge weights. Then, the network has a total weighted capacity of  $U := \sum_{\{i,j\} \in E} \mu_{ij} u_{ij}$ . Sending one unit of flow from  $O(k)$  to  $D(k)$  along a path  $p \in \mathcal{P}_k$  “consumes” a weighted capacity of  $\sum_{\{i,j\} \in p} \mu_{ij} u_{ij}$  and there exists a feasible multicommodity flow if all demands can be sent while consuming at most the total weighted capacity  $U$ . But how much weighted capacity

do we need to consume at least in order to send all the demands? The best we can do is to send  $d^k$  units along a shortest  $O(k)$ - $D(k)$ -path for all  $k \in \mathcal{K}$  and this consumes a weighted capacity of exactly  $\sum_{k \in \mathcal{K}} d^k \text{dist}_\mu(O(k), D(k))$ . Thus, if  $\sum_{\{i,j\} \in E} \mu_{ij} u_{ij} < \sum_{k \in \mathcal{K}} d^k \text{dist}_\mu(O(k), D(k))$  for any metric  $\mu$ , then no feasible multicommodity flow can exist under our choice of  $u$ . A rigorous proof of both the sufficiency and the necessity of the condition follows from applying Farkas' Lemma to the path formulation (2.32)–(2.36) in Chap. 2, Sect. 3.

The Japanese Theorem directly leads to the following capacity formulation for the MSCCRND problem with dynamic routing.

$$\text{Minimize } \sum_{\{i,j\} \in \mathcal{E}} c_{ij} u_{ij} \tag{11.100}$$

Subject to

$$\sum_{\{i,j\} \in \mathcal{E}} \mu_{ij} u_{ij} \geq \max_{d \in \mathcal{U}} \sum_{k \in \mathcal{K}} d^k \cdot \text{dist}_\mu(O(k), D(k)) \quad \forall \mu \in \mathcal{M} \tag{11.101}$$

$$u_{ij} \in \mathbf{Z}_{\geq 0} \quad \forall \{i, j\} \in E \tag{11.102}$$

The inequalities (11.101) are called *metric inequalities* and while there is an infinite number of metrics  $\mu \in \mathcal{M}$ , it is sufficient to include metric inequalities only for the (finitely many) extreme rays of the metric cone  $\mathcal{M}$ . The result is a finite integer linear program, but to make it practically viable, we need a separation algorithm. The only known separation algorithm for the metric inequalities so far is to write the separation problem as a bi-level (continuous) linear program and to then apply a standard transformation to turn it into a single-level quadratic program with complementary slackness conditions. This program can then be linearized; however, the linearization requires additional integer variables and big- $M$  constraints.

For static routing and budget uncertainty set  $\mathcal{U}_B(\bar{d}, \hat{d}, \Gamma)$ , a straight-forward generalization of the metric inequalities is not sufficient. In Claßen et al. (2015) the correct right hand side of (11.101) for this case is derived alongside with a polynomial time separation algorithm.

### 5.7 Strengthening the Formulations

The correctness of all the above formulations does not imply that the linear relaxation is close to an optimal integer solution. To improve the performance of branch-and-bound based solvers, the formulations can be strengthened with (facet-defining) valid inequalities. For this, the metric inequalities (11.101) are of particular interest. These inequalities are valid for the earlier formulations and connect capacities of different edges. Let us define the cut-metric

$$\mu_{ij} := \begin{cases} 1, & \text{if } i \in \mathcal{S} \text{ and } j \in \bar{\mathcal{S}} \\ 0, & \text{otherwise} \end{cases} \quad (11.103)$$

for some cut-set  $\mathcal{S} \subseteq V$ , i.e., the edges between  $\mathcal{S}$  and  $\bar{\mathcal{S}}$  have length 1 whereas all other edges have length 0. Clearly, the cut-metric is a metric, and hence the robust cut-set inequality

$$\sum_{\{i,j\} \in (\mathcal{S}, \bar{\mathcal{S}})} u_{ij} \geq \left\lceil \max_{d \in \mathcal{W}} \left( \sum_{k \in \mathcal{K}: O(k) \in \mathcal{S}, D(k) \in \bar{\mathcal{S}}} d^k + \sum_{k \in \mathcal{K}: O(k) \in \bar{\mathcal{S}}, D(k) \in \mathcal{S}} d^k \right) \right\rceil \quad (11.104)$$

is a valid inequality (since the left hand side is integer-valued, the right hand side can be rounded up to the next integer). For budget uncertainty, the right hand side can be computed by selecting the  $\Gamma$  largest deviations among those commodities crossing the cut (in addition to the nominal values). In fact, in this case the robust cut-set inequality (11.104) define a facet if both  $\mathcal{S}$  and  $\bar{\mathcal{S}}$  induce connected subgraphs and actual rounding is performed.

Further valid inequalities can be derived by considering  $k$ -partitions of the node set (cf. Sect. 4.4) or considering a single edge capacity constraint (11.47) (generalizing the so-called arc-residual capacity constraints, cf. Kutschka (2013)).

## 6 Bibliographical Notes

Robust optimization is an emerging field of research. Probably the earliest work in the field was reported by Soyster (1973). In the context of discrete optimization, several contribution were made by Kouvelis and Yu (1997). The budget uncertainty set was introduced by Bertsimas and Sim (2003, 2004) and is probably the most successful approach to date. A generalization of the budget uncertainty model to be used in the context of network design has been introduced by Büsing and D’Andreagiovanni (2012). We refer to Ben-Tal et al. (2009) for a robust optimization in continuous optimization. Multi-stage robustness was proposed by Ben-Tal et al. (2004). See Liebchen et al. (2009) for an introduction to recoverable robustness.

The first works for the single-commodity case of robust network design discuss the case where the uncertainty set is a finite list of scenarios. This case was first studied by Minoux (1989) and by Sanità (2009). Buchheim et al. (2011) propose the arc-flow-based formulation. Target cuts are due to Buchheim et al. (2008). Álvarez-Miranda et al. (2012) introduce the cut-set-based formulation with a separation algorithm for discrete scenario sets. The separation for cut-set inequalities under Hose uncertainty is due to Cacchiani et al. (2016). The 3-partition inequalities and their facet-defining properties were studied by Magnanti et al. (1993), Agarwal (2006), Cacchiani et al. (2016), and Schmidt (2014). Pesenti et al. (2004) study the network containment problem.



In the context of multicommodity network design, robust optimization was merely applied to communication networks. Belotti et al. (2008) consider network engineering. Altin et al. (2011) study network design with under the Hose uncertainty model and we refer to Koster et al. (2013) for network design under budget uncertainty.

The arc-flow based formulation for the MSCCRND problem with static routing and budget uncertainty is due to Koster et al. (2013); this includes the reformulation as a linear program and the separation algorithm. Altin et al. (2007) robustify the arc-flow formulation with Hose uncertainties. The path-flow formulation together with the solution algorithm was proposed by Ben-Ameur and Kerivin (2005).

The separation of robust metric inequalities for dynamic routing is due to Mattia (2013). Claßen et al. (2015) derive robust metric inequalities for static routing; the derivation of robust cutset inequalities as Chvátal-Gomory cuts is due to Koster et al. (2013).

The polynomial time algorithm for optimum tree routing templates for the VPN problem with unsplittable flows was given by Gupta et al. (2001). Goyal et al. (2008) prove that the VPN problem with unsplittable flows always has an optimum tree routing template.

For multi-stage robustness and affine recourse models in network design we refer to Atamtürk and Zhang (2007), Ben-Ameur (2007), and Poss and Raack (2013).

Instances of robust network design can be found at SNDlib Orłowski et al. (2010).

## 7 Conclusions and Perspectives

In recent years, network design has been on the one hand a fruitful application area for the emerging field of robust optimization. But, on the other hand, the robust network design has also been stimulating the further development of the robust optimization methodology. It can be expected that both these developments will continue in the years to come. In particular, multi-stage robustness concepts are still in their development and different applications will require new angles of view to arrive at suitable solutions. The increasing complexity by robustness concepts in general, and multi-stage concepts in particular, will force the development of new algorithmic ideas to deal with them.

## References

- Agarwal, Y. K. (2006). k-Partition-based facets of the network design problem. *Networks*, 47(3), 123–139.
- Altin, A., Amaldi, E., Belotti, P., & Pinar, M. Ç. (2007). Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1), 100–155.

- Altin, A., Yaman, H., & Pinar, M. (2011). The robust network loading problem under hose demand uncertainty: formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, 23(1), 75–89.
- Álvarez-Miranda, E., Cacchiani, V., Dorneth, T., Jünger, M., Liers, F., Lodi, A., et al. (2012). Models and algorithms for robust network design with several traffic scenarios. In A. Ridha Mahjoub, V. Markakis, I. Milis, & V. T. Paschos (Eds.), *ISCO 2012, Revised Selected Papers. Lecture notes in computer science* (vol. 7422, pp. 261–272). Springer.
- Atamtürk, A., & Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4), 662–673.
- Belotti, P., Capone, A., Carello, G., & Malucelli, F. (2008). Multi-layer mpls network design: The impact of statistical multiplexing. *Computer Networks*, 52(6), 1291–1307.
- Ben-Ameur, W. (2007). Between fully dynamic routing and robust stable routing. In *6th International Workshop on Design of Reliable Communication Networks, 2007, DRCN 2007* (pp. 1–6). IEEE
- Ben-Ameur, W., & Kerivin, H. (2005). Routing of uncertain demands. *Optimization and Engineering*, 3, 283–313.
- Ben-Tal, A., & Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1), 1–13.
- Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2), 351–376.
- Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust optimization*. Princeton: Princeton University Press
- Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1), 49–71.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53.
- Buchheim, C., Liers, F., & Oswald, M. (2008). Local cuts revisited. *Operations Research Letters*, 36(4), 430–433.
- Buchheim, C., Liers, F., & Sanità, L. (2011). An exact algorithm for robust network design. In J. Pahl, T. Reinert, & S. Voß (Eds.) *Proceedings of the INOC, INOC'11* (pp. 7–17). Springer.
- Büsing, C., & D'Andreagiovanni, F. (2012). New results about multi-band uncertainty in robust optimization. In *Proceedings Experimental Algorithms - 11th International Symposium, SEA 2012, Bordeaux, France, June 7–9, 2012*, pp. 63–74.
- Cacchiani, V., Jünger, M., Liers, F., Lodi, A., & Schmidt, D. R. (2016). Single-commodity robust network design with finite and Hose demand sets. *Mathematical Programming*, 157(1), 297–342.
- Claßen, G., Koster, A. M. C. A., Coudert, D., & Nepomuceno, N. (2014). Chance-constrained optimization of reliable fixed broadband wireless networks. *INFORMS Journal on Computing*, 26(4), 893–909.
- Claßen, G., Koster, A. M. C. A., Kutschka, M., & Tahiri, I. (2015). Robust metric inequalities for network loading under demand uncertainty. *Asia-Pacific Journal of Operational Research*, 32(5), 1550038, 1–27
- Duffield, N. G., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan, K. K., & van der Merwe, J. E. (1999). A flexible model for resource management in virtual private networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '99* (pp. 95–108). ACM Press.
- Fingerhut, J. A., Suri, S., & Turner, J. S. (1997). Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2), 287–309.
- Goyal, N., Olver, N., & Shepherd, B. (2008). The VPN conjecture is true. *Proceedings of the STOC*, 443–450.
- Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R., & Yener, B. (2001). Provisioning a virtual private network: a network design problem for multicommodity flow. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC '01* (pp. 389–398). ACM.
- Koster, A. M. C. A., Kutschka, M., & Raack, C. (2013). Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2), 128–149.

- Kouvelis, P., & Yu, G. (1997). *Robust discrete optimization and its applications*. Norwell, MA: Kluwer Academics Publishers.
- Kutschka, M. (2013). Robustness concepts for knapsack and network design problems under data uncertainty. Ph.D. thesis, RWTH Aachen University, <https://cu villier.de/de/shop/publications/6558>.
- Liebchen, C., Lübbecke, M., Möhring, R., & Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In R. K. Ahuja, R. H. Möhring, & C. D. Zaroliagis (Eds.) *Robust and online large-scale optimization: models and techniques for transportation systems* (pp 1–27). Springer.
- Magnanti, T. L., Mirchandani, P., & Vachani, R. (1993). The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60(1–3), 233–250.
- Mattia, S. (2013). The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, 54, 619–643.
- Minoux, M. (1989). Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3), 313–360.
- Onaga, K., & Kakusho, O. (1971). On feasibility conditions of multicommodity flows in networks. *Transactions on Circuit Theory*, 18(4), 425–429.
- Orlowski, S., Wessälly, R., Pióro, M., & Tomaszewski, A. (2010). SNDlib 1.0—survivable network design library. *Networks*, 55(3), 276–286.
- Pascali, F. (2009). Chance constrained network design. Ph.D. thesis, University of Pisa. <https://etd.adm.unipi.it/etd-11262009-005543/>.
- Pesenti, R., Rinaldi, F., & Ukovich, W. (2004). An exact algorithm for the min-cost network containment problem. *Networks*, 43(2), 87–102.
- Poss, M., & Raack, C. (2013). Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2), 150–155.
- Sanità, L. (2009). Robust network design. Ph.D. thesis, Università La Sapienza, Roma.
- Schmidt, D. R. (2014). Robust design of single-commodity networks. Ph.D. thesis, Universität zu Köln.
- Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21, 1154–1157.