A Comparison of Data-Driven Uncertainty Sets for Robust Network Design

Francis Garuba\(^1\), Marc Goerigk\(^2\), and Peter Jacko\(^1\)

\(^1\)Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom
\(^2\)Network and Data Science Management, University of Siegen, Unterer Schloß 3, 57072 Siegen, Germany

Abstract

We consider a network design and expansion problem, where we need to make a capacity investment now, such that uncertain future demand can be satisfied as closely as possible. To use a robust optimization approach, we need to construct an uncertainty set that contains all scenarios that we believe to be possible. In this paper we discuss how to actually construct two common models of uncertainty set, discrete and polyhedral uncertainty, using data-driven techniques on real-world data. We employ clustering to generate a discrete uncertainty set, and supervised learning to generate a polyhedral uncertainty set. We then compare the performance of the resulting robust solutions for these two types of models on real-world data. Our results indicate that polyhedral models, while being popular in the recent literature, are less effective than discrete models both in terms of computational burden and solution quality regardless of the performance measure considered (worst-case, conditional value-at-risk, average).

Keywords: network design; robust optimization; optimization in telecommunications; data-driven optimization; clustering; supervised learning

1 Introduction

Operations research approaches have found wide application in the planning, design and operations management of transportation, power and energy distribution, supply chain logistics and telecommunications networks. In particular, many types of optimization models have been developed over the last decades for network design and expansion problems, see, e.g., [MW84, Min89, Ber98].

In telecommunications, for instance, network design models can be used to curb congestion and to provide an acceptable quality of service to the subscribers. Effort to provide an acceptable service has resulted in capital expenditure of billions of USD in global telecoms investment. Optimization of investments has thus attained a key strategic role in this industry. Moreover, these decisions need to be made well ahead of time based on a forecast of future traffic demand.

Unfortunately, traffic demand has proven to be difficult to predict accurately. In order to factor in this uncertainty and design a network that is immune to traffic variability, robust optimization approaches have been proposed. For this purpose, a number of uncertainty models.
have already been developed and investigated (see [GS16, BTEGN09, BBC11]). The drawback of classic approaches, however, is that the uncertainty set is assumed to be given, i.e., the decision maker can advise on how the uncertainty is shaped. Moreover, an inappropriate choice of uncertainty set may result in models that are too conservative or in some cases computationally intractable. As the decision maker cannot be expected to make this choice in practice, data-driven and learning approaches have been proposed (see [BGK17, CDG19]).

This paper contributes to this recent line of research proposing a clustering approach to generate discrete uncertainty sets from real data viewed as a set of scenarios. We use the $K$-means clustering method which results in aggregating similar scenarios into clusters and representing each cluster of scenarios by its centroid, with the intention to reduce the problem complexity on the one hand, and to become less dependent on data noise on the other hand.

The basic network design problem that we consider in this paper is as follows. Given an undirected graph $G = (V, E)$ and currently installed capacity $u_e$ for each edge $e \in E$, we would like to determine an amount of capacity $x_e$ to be installed additionally. For each edge $e$, we are given investment cost $c_e$ per unit of additionally installed capacity. As the graph is undirected, the direction flow is not relevant for our model, and we define $K = \{\{i, j\} : i, j \in V, i < j\}$ as the set of commodities, where each commodity $k$ is identified by an unordered pair of nodes $\{i, j\}$ between which a given demand needs to be satisfied. Let $d_k$ be the demand corresponding to commodity $k \in K$, and let $P_k$ be the set of simple paths in $G$ connecting the nodes of the commodity. The aim is to find capacities $x$ such that all demands are fulfilled and the capacity expansion costs are as small as possible. Formally, the baseline model can thus be written as follows.

\[
\begin{align*}
\min & \sum_{e \in E} c_e x_e \\
\text{s.t.} & \sum_{p \in P_k} f_{kp} \geq d_k \quad \forall k \in K \\
& \sum_{k \in K} \sum_{p \in P_k, e \in p} f_{kp} \leq u_e + x_e \quad \forall e \in E \\
& x_e \geq 0 \quad \forall e \in E \\
& f_{kp} \geq 0 \quad \forall k \in K, p \in P_k
\end{align*}
\]

The variables $f_{kp}$ model the amount of flow along path $p$ for commodity $k$. Here, Constraints (2) ensure that a sufficient amount of flow is transported along all paths connecting source and sink of commodity $k \in K$, while Constraints (3) model that each edge needs to provide sufficient capacity. Instead of using a path-based formulation, it is also possible to use a model with flow variables for every edge in the network (see, e.g., [GCF99]). In this paper, we focus on the path-based formulation, as it performed better in our computational experiments.

In practice, the demand $d$ changes over time and is not known precisely. Thus, a two-stage model is required, where we decide now where to build how much capacity (the strategic decision $x$), and we can decide where to route the flow once the demand is known (the operational decision $f$). Let us assume that a set $U$ can be constructed that contains all demand scenarios $d$ that we would like to take into account for our planning. The two-stage robust network design problem is then to solve

\[
\begin{align*}
\min & \sum_{e \in E} c_e x_e \\
\text{s.t.} & \sum_{p \in P_k} f_{kp}(d) \geq d_k \quad \forall k \in K, d \in U
\end{align*}
\]
In this setting, $f_{kp}$ has become a function that depends on the scenario $d$. Note that in Constraint (7) $d_k$ is a component of $d$, thus is also scenario-dependent.

Robust optimization in general has found increasing use and application in the network design area. [AZ07] considered a two-stage robust network flow problem under demand uncertainty following the work of [BTGGN04], while [OV07] introduced affine routing in the their robust network capacity planning model. [OZ07] looked at network capacity expansion under both demand and cost uncertainty. [KKR13] considered a robust network design problem with static routing in the setting of [BS04]. [PR12] considered robust network design with polyhedral uncertainty and [BVKO13] robust capacity assignment for networks with uncertain demand. [PP15] used a cutting plane algorithm while taking into consideration the uncertainty in unmet demand outsourced cost.

Regarding uncertainty sets, polyhedral models are most frequently used in radio network design, along with hose models from the works of [DGG+99, FST97], budget uncertainty by [AZ07], cardinal constrained uncertainty by [BS04], and interval uncertainty, among others. Little research compares these models of uncertainty. [AZ07] compared their single-stage robust model using budget uncertainty with a scenario-based two-stage stochastic approach. [CDG19] constructed different uncertainty sets from real world data and compared performance within and outside sample for shortest path problems.

In this paper we present the following contributions:

- We propose and develop a clustering approach (using the well-known $K$-means clustering data mining method) to generate discrete uncertainty sets from real data for a network design and network expansion problems;

- We use this approach to calculate the cluster centroids for real-world data taken from SNDlib (see [OWPT10]) and use these centroids to define a discrete uncertainty set which is used to compute an optimal network expansion;

- We compare this solution to the solution obtained using the state-of-the-art approach of modelling uncertainty using a polyhedral set, where constraints on the demand are given as hyperplanes generated dynamically using supervised learning. To the best of our knowledge, this is for the first time that such a comparison is done for network design problems;

- For the real-world dataset we consider, we find in our numerical experiments that solutions based on discrete uncertainty found by clustering outperform solutions based on polyhedral uncertainty found by supervised learning when using high risk-adverse performance metrics such as maximum or CVaR$^{0.95}$ of unsatisfied demand. This is less clear for less risk-averse metrics such as expected value or CVaR$^{0.75}$ of unsatisfied demand, but the clustering approach is still superior in most cases. At the same time, solutions based on discrete uncertainty found by clustering can be computed two orders of magnitude faster than those based on polyhedral uncertainty found by supervised learning.

The rest of this paper is organized as follows. As the problem data is the center point of our research, we first discuss this in Section 2. In particular, we describe how to construct uncertainty sets $\mathcal{U}$ from the data. We then introduce models for robust network design for
both discrete and polyhedral uncertainty sets in Section 3. Experimental results are discussed in Section 4. Finally, Section 5 concludes our work and points out future research directions.

2 Problem Data and Uncertainty Set Construction

We focus on the Abilene network based on data from the SNDlib (see [OWPT10]). It consists of 12 nodes connected by 15 edges, see Figure 1, which spread over the US. With 12 nodes, there exist \(12 \cdot \frac{11}{2} = 66 =: \kappa\) different commodities.

Data was collected by Yin Zhang\(^1\) in 5 minute intervals between 01.03.2004 and 10.09.2004 with some breaks in between. Table 1 shows the number of measurements that are available for each month. Note that one day can give 288 measurements in 5 minute intervals. Based on this number, we also show the maximum number of possible measurements that can be achieved each month, but note that not all data is available.

![Abilene network topology.](image)

| Month | 03   | 04   | 05   | 06   | 07   | 08   | 09   |
|-------|------|------|------|------|------|------|------|
| # Measurements available | 4,032 | 6,048 | 8,928 | 8,640 | 8,928 | 8,640 | 2,880 |
| # Measurements possible   | 8,928 | 8,640 | 8,928 | 8,640 | 8,928 | 8,928 | 8,640 |

Table 1: Numbers of available measurements for each month.

We use an arbitrary set of \(T\) measurements for the purpose of model training. The rest of the data is then used for the evaluation of results. This means the training data consists of \(T\) demand scenarios, each being a \(\kappa\)-dimensional vector of reals. The total demand per scenario, i.e., the sum of demand over all commodities, for months May-August which provide the most complete sets of data measurements, is presented in Figure 2.

We now discuss how to generate discrete and polyhedral uncertainty sets based on the training data points. Let \(\mathcal{D} = \{d^1, \ldots, d^T\}\) denote this training set. For a discrete uncertainty set \(\mathcal{U}_d\), where each scenario is explicitly listed, a natural approach is setting \(\mathcal{U}_d = \mathcal{D}\). But it has been shown (see [CDG19]) that this can lead to an overfitting effect, such that the resulting robust solutions do not perform well on out-of-sample data points. Furthermore, it is desirable to control the degree of conservatism. We therefore propose a clustering approach to generate discrete uncertainty sets. We aggregate similar scenarios together, with the intention to reduce the problem complexity on the one hand, and to become less dependent on data noise on the other hand. Scenario aggregation based on \(K\)-means clustering has been applied as an approximation method also to robust min-knapsack problems, see [CGKZ18]. Let \(\mathcal{U}_d^K\) denote a discrete uncertainty set derived from a \(K\)-means clustering of the set \(\mathcal{D}\). Then on the one

\(^1\)http://www.cs.utexas.edu/~yzhang/
(a) Demand profile for month 05.

(b) Demand profile for month 06.

(c) Demand profile for month 07 (training set).

(d) Demand profile for month 08.

Figure 2: Total demand in the Abilene network.
boundary case, $U_d^1 = D$, i.e., we contain the original set of training points as a special case, and on the other boundary case, $U_d^1$ consists of only the average case scenario.

We show a simple example in Figure 3. In Figure 3a, we plot a subset of the training data, restricted to two arbitrarily chosen dimensions (recall that every commodity corresponds to a dimension of the demand vector). In Figure 3c, we show a discrete uncertainty $U_d^5$ set based on a $K$-means clustering with $K = 5$ centers, which captures the training data only in a rough manner. With $K = 20$ (see Figure 3e), most features of the data have been captured.

We now consider the case of polyhedral uncertainty,

$$U_p^M = \{ \mathbf{d} \in \mathbb{R}_{+}^{\kappa} : V \mathbf{d} \leq \mathbf{b}, d_k \in [d_k, \bar{d}_k] \}$$

where $V = (v_{ik})$ is a matrix in $\mathbb{R}^{M \times \kappa}$ and $\mathbf{b}$ is a vector in $\mathbb{R}^\kappa$ (i.e., there are $M$ linear constraints on the demand vector). As the number of constraints $M$ will have a significant impact on the solution time of the resulting robust model, we would like to find only few constraints which describe the training data $D$ well. To this end, we apply a technique similar to supervised learning in machine learning. We generate a set of noise data points, which we would like to distinguish from the original training data by placing hyperplanes that put as many original points on one side, and as many noise points on the other side as possible. This trade-off is adjusted dynamically: for the first hyperplane, there is a high penalty for original points that are classified as noise. This way, we find an outer description of the data, which results in large and conservative uncertainty sets. This penalty is reduced over time, so that later hyperplanes become less conservative and cut away outliers in the training data. Noise points are generated by randomly increasing values of single training data points, and randomly using values from other data points in single dimensions with low probability.

In Figure 3, we use the same data as for the clustering example to illustrate this process. The random noise is shown as red points in Figure 3b. The first four hyperplanes we generate are given in Figure 3d. It can be seen that they form an outer approximation of the data, removing only few outliers in the process. With an increasing number of hyperplanes $M$, the polyhedron $U_p^M$ becomes smaller and less conservative (see Figure 3f).

3 Robust Models

We now discuss how to reformulate the general model (6-10) for specific uncertainty sets $U_d^d$ and $U_p^M$.

3.1 Discrete Uncertainty

Let $U_K^d = \{ \mathbf{d}^1, \ldots, \mathbf{d}^K \}$ be given. As this set is discrete, we can simply write $f_{kp}(\mathbf{d}) = f_{kp}^i$ for all $\mathbf{d} = \mathbf{d}^i \in U_K^d$. We write $|K| := \{1, \ldots, K\}$ in the following. The resulting compact optimization model is then given as follows:

$$\min \sum_{e \in \mathcal{E}} c_e x_e$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_k} f_{kp} \geq d_k^i \quad \forall k \in \mathcal{K}, i \in [K]$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k} e \in \mathcal{E}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k} f_{kp} \leq u_e + x_e \quad \forall e \in \mathcal{E}, i \in [K]$$

$$x_e \geq 0 \quad \forall e \in \mathcal{E}$$

$$f_{kp} \geq 0 \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, i \in [K]$$
Figure 3: Illustration of methods to generate discrete and polyhedral uncertainty on a subset of training data restricted to two dimensions (note the logarithmic scale of the axes).
3.2 Polyhedral Uncertainty

Rewriting \( f_{kp}(d) \) in a compact form is less straightforward for continuous uncertainty sets than in the previous, discrete case. We apply the well-known affine decision rules (also known as affine adjustable robust counterpart) approach, see [BTGGN04]. To this end, we restrict \( f_{kp}(d) \) to be an affine linear function in \( d \) by writing

\[
f_{kp}(d) = \phi_{kp} + \sum_{\ell \in K} d_{\ell} \Phi_{k\ell}
\]

Here, \( \phi_{kp} \geq 0 \) and \( \Phi_{k\ell} \geq 0 \) are new decision variables. By using affine decision rules, we restrict the set of feasible solutions, and thus form a conservative approximation to the original problem.

By substituting the \( f_{kp}(d) \) variables in (6-10) and rearranging terms, the problem becomes:

\[
\begin{align*}
\min \sum_{e \in E} c_{e} x_{e} & \quad \text{(16)} \\
\sum_{p \in P_{k}} \phi_{kp} \geq \max_{d \in U_{p}^{k}} \sum_{\ell \in K} \left( 1_{\ell = k} - \sum_{p \in P_{k}} \Phi_{kp\ell} \right) d_{\ell} & \quad \forall k \in K \quad (17) \\
\sum_{k \in K} \sum_{p \in P_{k}, e \in p} \phi_{kp} + \max_{d \in U_{p}^{k}} \sum_{\ell \in K} \left( \sum_{k \in K} \sum_{p \in P_{k}, e \in p} \Phi_{kp\ell} \right) d_{\ell}^{e} & \leq u_{e} + x_{e} \quad \forall e \in E \quad (18) \\
\phi_{kp} + \min_{d \in U_{p}^{k}} \sum_{\ell \in K} \Phi_{kp\ell} d_{\ell} & \geq 0 \quad \forall k \in K, p \in P_{k} \quad (19) \\
x_{e} & \geq 0 \quad \forall e \in E \quad (20) \\
\phi_{kp} & \geq 0 \quad \forall k \in K, p \in P_{k} \quad (21) \\
\Phi_{k\ell} & \geq 0 \quad \forall k \in K, p \in P_{k}, \ell \in K \quad (22)
\end{align*}
\]

The inner maximization and minimization problems can then be reformulated using linear programming duality. As an example, consider Constraint (17) for a fixed \( k \in K \). The value of the right-hand side is

\[
\max \sum_{\ell \in K} \left( 1_{\ell = k} - \sum_{p \in P_{k}} \Phi_{kp\ell} \right) d_{\ell} \quad (23)
\]

\[
\text{s.t. } V d \leq b \quad (24)
\]

\[
d_{\ell} \in [d_{\ell}, \tilde{d}_{\ell}] \quad \forall \ell \in K \quad (25)
\]

As \( U_{p}^{k} \) is polyhedral, this is a linear program, the dual of which is

\[
\begin{align*}
\min \sum_{i \in [M]} b_{i} \alpha_{i} + \sum_{\ell \in K} (\tilde{d}_{\ell} \beta_{\ell} - d_{\ell} \hat{\beta}_{\ell}) & \quad (26) \\
\text{s.t. } \sum_{i \in [M]} v_{i} \alpha_{i} + \beta_{\ell} \geq 1_{\ell = k} - \sum_{p \in P_{k}} \Phi_{kp\ell} & \quad \forall \ell \in K \quad (27) \\
\alpha, \beta, \hat{\beta} & \geq 0 \quad (28)
\end{align*}
\]

By weak duality, any feasible solution to (26-28) gives an upper bound on the value of (23-25). Thus we can substitute the formulation (26-28) into Constraint (17) to reach an equivalent
linear reformulation. Repeating this process for all constraints, the robust network extension problem with polyhedral uncertainty can be rewritten in the following way:

\[
\begin{align*}
\min & \sum_{e \in \mathcal{E}} c_e x_e \\
\text{s.t.} & \sum_{p \in \mathcal{P}_k} \phi_{kp} \geq \sum_{i \in [M]} b_i \alpha_{ki} + \sum_{\ell \in \mathcal{K}} (d_\ell \beta_{kl} - d_\ell \beta_{kl}) \quad \forall k \in \mathcal{K} \\
& \sum_{i \in [M]} v_{i \ell} \alpha_{ki} + \beta_{kl} \geq 1_{\ell=k} - \sum_{p \in \mathcal{P}_k} \phi_{kp} \quad \forall k, \ell \in \mathcal{K} \\
& \sum_{i \in [M]} \sum_{p \in \mathcal{P}_k, e \in \mathcal{E}} \phi_{kp} + \sum_{i \in [M]} (d_i \pi_{ei} - d_i \pi_{ei}) \leq u_e + x_e \quad \forall e \in \mathcal{E} \\
& \sum_{i \in [M]} \sum_{p \in \mathcal{P}_k, e \in \mathcal{E}} \phi_{kp} + \sum_{i \in [M]} (d_i \pi_{ei} - d_i \pi_{ei}) \geq \sum_{k \in \mathcal{K}, p \in \mathcal{P}_k, e \in \mathcal{E}} \Phi_{kp} \quad \forall e \in \mathcal{E}, \ell \in \mathcal{K} \\
& \phi_{kp} \geq \sum_{i \in [M]} b_i \xi_{kpi} + \sum_{\ell \in \mathcal{K}} (d_i \zeta_{kpe} - d_i \zeta_{kpe}) \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k \\
& \sum_{i \in [M]} \sum_{p \in \mathcal{P}_k} \xi_{kpi} + \zeta_{kpe} - \zeta_{kpe} \geq -\Phi_{kp} \quad \forall k, \ell \in \mathcal{K}, p \in \mathcal{P}_k \\
& x_e \geq 0 \quad \forall e \in \mathcal{E} \\
& \phi_{kp} \geq 0 \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k \\
& \Phi_{kp} \geq 0 \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, \ell \in \mathcal{K} \\
& \alpha_{ki} \geq 0 \quad \forall i \in [M], k \in \mathcal{K} \\
& \beta_{kl} \geq 0 \quad \forall k, \ell \in \mathcal{K} \\
& \pi_{ei} \geq 0 \quad \forall e \in \mathcal{E}, i \in [M] \\
& \pi_{ei} \geq 0 \quad \forall e \in \mathcal{E}, \ell \in \mathcal{K} \\
& \xi_{kpi} \geq 0 \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, i \in [M] \\
& \zeta_{kpe} \geq 0 \quad \forall k, \ell \in \mathcal{K}, p \in \mathcal{P}_k
\end{align*}
\]

4 Experiments

4.1 Setup

The aim of the experiments is to analyze the performance of solutions to the robust network design problem using discrete and polyhedral uncertainty sets, respectively. We set all existing capacities \( u_e \) to be zero, so that the effect of model choice becomes more visible.

Using the data described in Section 2, we focus on the four months from beginning of May until end of August which provide the most complete sets of data measurements. We based the training set on an arbitrarily chosen month, 07, which consists of 8,928 demand scenarios, but we removed outlier scenarios, which are defined as the top 2% of scenarios with regard to total demand, leaving us with \( T = 8,750 \) scenarios in the training set. The corresponding cut-off value is shown in Figure 2c as a horizontal blue line.

We calculate solutions based on the training set derived from month 07 measurements and then evaluate them on all the scenarios from the training set and from the three other months (05, 06, 08), minimizing unsatisfied demand. Only the first-stage \( \mathbf{x} \)-part of a solution is used for evaluation.
For discrete uncertainty, we calculate solutions based on clusterings with \( K = 100 \) up to \( K = 8,600 \) in steps of 100, and in addition using all \( T = 8,750 \) training scenarios (a total of 87 optimization problems and solutions). Clusters are calculated using the \texttt{kmeans} function of SciPy 1.2.1 under Python 3.7.

For polyhedral uncertainty, we placed hyperplanes using the \texttt{dual_annealing} function from SciPy. We generate 140 hyperplanes this way. They are collected in 28 polyhedra, where polyhedron \( i \) uses all hyperplanes of polyhedron \( i - 1 \), and five more in the order that they were generated. In total, this means that 28 optimization problems with polyhedral uncertainty are solved.

For a better comparison, the \( 87 + 28 \) solutions are then also scaled up and down uniformly by multiplying the corresponding \( x \) vector with a factor \( \lambda = 0.5 \) up to \( \lambda = 1.5 \) (with step size \( 1/40 \)). The reason to also consider these scaled versions is because, by construction, solutions based on polyhedra will be more conservative than those based on clusterings. By scaling solutions up and down, a more comprehensive comparison becomes possible.

Each of these \( 87 + 28 \cdot 41 \) solutions is then evaluated by calculating an optimal flow for each of the 8750 training scenarios and each of the 8928 + 8640 + 8928 evaluation scenarios. In total, this means that over 166 million linear programs are solved for the evaluation. As there may not be sufficient capacity available to route all demand, we minimize the unsatisfied demand in each optimization problem. The corresponding model to evaluate solutions \( x \) for a fixed scenario \( d \) is as follows:

\[
\begin{align*}
\min & \sum_{k \in \mathcal{K}} h_k \\
\text{s.t.} & \quad h_k \geq d_k - \sum_{p \in \mathcal{P}_k} f_{kp} \quad \forall k \in \mathcal{K} \\
& \quad \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k, e \in p} f_{kp} \leq u_e + x_e \quad \forall e \in \mathcal{E} \\
& \quad x_e \geq 0 \quad \forall e \in \mathcal{E} \\
& \quad f_{kp} \geq 0 \quad \forall k \in \mathcal{K}, p \in \mathcal{P}_k
\end{align*}
\]

where \( h_k \) denotes the unsatisfied demand in commodity \( k \). Note that the cost of a solution only depends on the choice of \( x \). Additionally, for each month, we calculate four measures to characterize the performance of a solution with regard to unsatisfied demand: average, CVaR\(_{0.75}\) (i.e., the average unsatisfied demand over the 25% largest values), CVaR\(_{0.95}\), and the maximum.

### 4.2 Results

We first discuss the performance of solutions on the training set (month 07) in the left column of Figure 4. On the horizontal axis, we show the costs of solutions, while the vertical axis shows the four measures of unsatisfied demand. Each point corresponds to a solution (87 black squares corresponding to the discrete uncertainty solutions, 28 blue crosses corresponding to the polyhedral uncertainty solutions). The lines show the performance of the scaled solutions.

Consider Figure 4g. By construction, we know that the discrete uncertainty solution with \( U_{d,0750} \) has zero unsatisfied demand on the training set, and is the cheapest possible solution to do so. Most polyhedral solutions use conservative outer approximations of the training data and thus also have zero unmet demand, but at higher costs. We can also see that solutions

\[\text{2All linear programs were solved using Cplex 12.8 on a virtual Ubuntu server with ten Xeon CPU E7-2850 processors at 2.00 GHz speed and 23.5 GB RAM using only one core each.}\]
that use fewer clusters or more hyperplanes become less conservative, allowing unsatisfied demand at lower solution costs. The behaviour we see in Figure 4g is to be expected by design. The open question is whether it can also be observed on evaluation data.

Figures 4a, 4c and 4e show the average, CVaR$_{0.75}$, and CVaR$_{0.95}$ performance on the training set, respectively. Here the differences between both types of solution are much less pronounced; we see that blue and black lines overlap, indicating a similar performance of solution types.

Compare this performance to the right-hand column of Figure 4, where the performance on month 08 is presented. The order of magnitude of unsatisfied demand has increased for each type of solution: whereas in Figure 4g we can reach zero unsatisfied demand, the same solutions have between five and seven thousand units of maximum unmet demand in Figure 4h. But the relative performance between the solution types is similar. Whereas solutions based on polyhedral uncertainty generally have a higher degree of robustness at higher investment costs, it is possible to scale solutions based on clustered data up to reach solutions with a similar degree of robustness at lower costs. This is particularly visible for the high risk-adverse measures in Figures 4f and 4h, whereas these performance differences are less clear-cut for the less risk-adverse measures in Figures 4b and 4d.

Figure 6 in Appendix A shows the results for months 05 and 06, where the same observations apply as for month 08.

In terms of solution quality, i.e., trade-off between investment costs and unsatisfied demand, we thus find the following result: Solutions based on discrete uncertainty found by clustering outperform solutions based on polyhedral uncertainty found by supervised learning when using high risk-adverse performance metrics such as maximum or CVaR$_{0.95}$ of unsatisfied demand. This is less clear for less risk-adverse metrics such as expected value or CVaR$_{0.75}$ of unsatisfied demand, but the clustering approach is still superior in most cases.

We now consider the time required to solve the corresponding robust optimization models. Figure 5 shows the Cplex solution time for discrete and for polyhedral uncertainty, which depends on the size of the uncertainty set (note the two different horizontal axes and the logarithmic vertical axis).

It can be seen that even the largest discrete model (that uses all training scenarios directly) is still easier to solve than the smallest polyhedral model (using five hyperplanes in addition to the lower and upper bounds). So this experiment reveals that using discrete uncertainty sets not only results in a better solution quality, they are also easier to solve.

5 Conclusions

In the robust optimization literature, frequently both discrete and polyhedral uncertainty sets are being used. In this paper we compared the resulting solutions using real-world data for a network expansion problem. We describe how to construct uncertainty sets based on clustering the training data using a well-known data mining technique, and by separating training data from noise using a well-known machine learning method. In our computational study we found that solutions based on discrete uncertainty models outperform solutions based on polyhedral models in most performance metrics, and are also easier to compute. The strong performance of discrete uncertainty sets is in line with evidence from the experiment on shortest path data performed in [CDG19]. This also indicates that the current network design literature, which has a strong focus on polyhedral models, may benefit from considering simple discrete models more.

One reason for this observation may be that the raw data itself does not have a convex shape, and thus an approximation by a convex polyhedron is inadequate. Potentially, a robust...
Figure 4: Results for training set and month 08.
optimization approach can use an uncertainty set \( \mathcal{U} \) that is the union of multiple polyhedra. While for one-stage min-max problems it holds that optimizing with respect to \( \mathcal{U} \) or its convex hull is equivalent, this is not the case for two-stage problems. Two-stage network design with an uncertainty set that is the union of polyhedra may therefore have the potential to reach better solutions than by using a single polyhedron as model for the uncertainty. However, such an approach will come at additional computational cost.

**References**

[AZ07] A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, August 2007.

[BBC11] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, January 2011.

[Ber98] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, Belmont, Massachusetts, 1998.

[BGK17] D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, Feb 2017.

[BS04] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, February 2004.

[BTEGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[BTGGN04] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, March 2004.

[BVKO13] F. Babonneau, J.-P. Vial, O. Klopfenstein, and A. Ouorou. Robust capacity assignment solutions for telecommunications networks with uncertain demands. *Networks*, 62(4):255–272, 2013.
A. Chassein, T. Dokka, and M. Goerigk. Algorithms and uncertainty sets for data-driven robust shortest path problems. *European Journal of Operational Research*, 274(2):671 – 686, 2019.

A. Chassein, M. Goerigk, A. Kasperski, and P. Zieliński. Approximating multiobjective combinatorial optimization problems with the OWA criterion. *arXiv preprint arXiv:1804.03594*, 2018.

N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive. A flexible model for resource management in virtual private networks. *ACM SIGCOMM Computer Communication Review*, 29(4):95–108, October 1999.

J. A. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309, August 1997.

Bernard Gendron, Teodor Gabriel Crainic, and Antonio Frangioni. Multicommodity capacitated network design. *Springer*, pages 1–19, 1999.

M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. In *Algorithm engineering*, pages 245–279. Springer International Publishing, 2016.

A. M. C. A. Koster, M. Kutschka, and C. Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, January 2013.

M. Minoux. Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360, 1989.

T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, February 1984.

A. Ouorou and J.-P. Vial. A model for robust capacity planning for telecommunications networks under demand uncertainty. In *6th International Workshop on Design and Reliable Communication Networks (DRCN 2007)*, pages 1–4. IEEE, 2007.

S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0-Survivable Network Design Library. *Networks*, 55(3):276–286, May 2010.

F. Ordóñez and J. Zhao. Robust capacity expansion of network flows. *Networks*, 50(2):136–145, 2007.

A. A. Pessoa and M. Poss. Robust network design with uncertain outsourcing cost. *INFORMS Journal on Computing*, 27(3):507–524, August 2015.

M. Poss and C. Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, September 2012.

### A Additional Experimental Results
Figure 6: Results for months 05 and 06.