Variable-Sized Uncertainty and Inverse Problems in Robust Optimization

André Chassein*[1] and Marc Goerigk†[2]

[1]Fachbereich Mathematik, Technische Universität Kaiserslautern, 67653 Kaiserslautern, Germany

[2]Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom

Abstract

In robust optimization, the general aim is to find a solution that performs well over a set of possible parameter outcomes, the so-called uncertainty set. In this paper, we assume that the uncertainty size is not fixed, and instead aim at finding a set of robust solutions that covers all possible uncertainty set outcomes. We refer to these problems as robust optimization with variable-sized uncertainty. We discuss how to construct smallest possible sets of min-max robust solutions and give bounds on their size.

A special case of this perspective is to analyze for which uncertainty sets a nominal solution ceases to be a robust solution, which amounts to an inverse robust optimization problem. We consider this problem with a min-max regret objective and present mixed-integer linear programming formulations that can be applied to construct suitable uncertainty sets.

Results on both variable-sized uncertainty and inverse problems are further supported with experimental data.

Keywords: Robustness and sensitivity analysis; uncertainty sets; inverse optimization; optimization under uncertainty

1 Introduction

Robust optimization has become a vibrant field of research with fruitful practical applications, of which several recent surveys give testimonial (see [ABV09, BTGN09, BBC11, GYd15, GS16, CG16b]). Two of the most widely used approaches to robust optimization are the so-called (absolute) min-max and min-max regret approaches (see, e.g., the

*Email: chassein@mathematik.uni-kl.de

†Corresponding author. Email: m.goerigk@lancaster.ac.uk

classic book on the topic [KY97]). For some combinatorial optimization problem of the form

$$(\mathsf{P}) \qquad \min\ \left\{c^t x : x \in \mathcal{X} \subseteq \{0,1\}^n\right\}$$

with a set of feasible solutions $\mathcal{X}$, let $\mathcal{U}$ denote the set of all possible scenario outcomes for the objective function parameter c. Then the min-max counterpart of the problem is given as

$$\min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}} c^t x$$

and the min-max regret counterpart is given as

$$\min_{x \in \mathcal{X}} reg(x, \mathcal{U})$$

with

$$reg(x, \mathcal{U}) := \max_{c \in \mathcal{U}} \left(c^t x - opt(c)\right)$$

where $opt(c)$ denotes the optimal objective value for problem (P) with objective c.

In the recent literature, the problem of finding suitable sets $\mathcal{U}$ has come to the focus of attention, see [BS04, BPS04, BB09]. This acknowledges that the set $\mathcal{U}$ might not be "given" by a real-world practitioner, but is part of the responsibility of the operations researcher.

In this paper we consider the question how robust solutions change when the size of the uncertainty set changes. We call this approach variable-sized robust optimization and analyze how to find minimal sets of robust solutions that can be applied to any possible uncertainty sets. This way, the decision maker is presented with candidate solutions that are robust for different-sized uncertainty sets, and he can choose which suits him best. Results on this approach for min-max robust optimization are discussed in Section 2.

The notion of *variable uncertainty* has also been used in [Pos13], but is different to our approach: In their paper, the size of the uncertainty set depends on the solution x, i.e., $\mathcal{U} = \mathcal{U}(x)$, while we use size parameter that does not depend on x. Further related is the notion of parametric programming (see, e.g., [Car83]). Our approach can be seen as being a parametric robust optimization problem.

As a special case of variable-sized robust optimization, we consider the following question: Given only a nominal problem (P) with objective $\hat{c}$, how large can an uncertainty set become, such that the nominal solution still remains optimal for the resulting robust problem? Due to the similarity in our question to inverse optimization problems, see, e.g., [AO01, Heu04, ABP09, NC15] we denote this as the inverse perspective to robust optimization. The approach from [CN03] is remotely related to our perspective. There, the authors define the robustness of a solution via the largest possible deviation of the problem coefficients in a facility location setting. In a similar spirit, the *R-model* for robust satisficing [JJNS16] aims at finding a solution that remains feasible for an uncertainty set that is as large as possible. Our approach can also be used as a means of sensitivity analysis. Given several solutions that are optimal for some nominal problem, the decision maker can choose one that is most robust in our sense. In the same vein, one can check which parts of a solution are particularly fragile, and strengthen them further. This approach is presented for min-max regret in Section 3.

Our paper closes with a conclusion and discussion of further research directions in Section 4.

2 Variable-Sized Min-Max Robust Optimization

In this section, we analyze how optimal robust solutions change when the size of the uncertainty set increases. We assume to have information about the midpoint (nominal) scenario $\hat{c}$, and the shape of the uncertainty set $\mathcal{U}$. The actual size of the uncertainty set is assumed to be uncertain.

More formally, we assume that the uncertainty set is given in the form

$$\mathcal{U}_\lambda = \{\hat{c}\} + \lambda B$$

where B is a convex set containing the origin and $\hat{c} > 0$ is the midpoint of the uncertainty set. The parameter $\lambda \geq 0$ is an indicator for the size of the uncertainty set. For $\lambda = 0$, we have $\mathcal{U}_0 = \{\hat{c}\}$, i.e., the nominal problem, and for $\lambda \to \infty$ we obtain the extreme case of complete uncertainty.

We consider the min-max robust optimization problem

$$\min_{x\in\mathcal{X}} \max_{c\in\mathcal{U}_\lambda} c^t x. \tag{$\mathsf{P}(\lambda)$}$$

The goal of variable-sized robust optimization is to compute a minimal set $\mathcal{S} \subset \mathcal{X}$ such that for any $\lambda \geq 0$, $\mathcal{S}$ contains a solution that is optimal for $\mathsf{P}(\lambda)$. Note that for $\lambda = 0$, set $\mathcal{S}$ must contain a solution $\hat{x}$ that is optimal for the nominal problem. By increasing λ, we trace how this solution needs to change with increasing degree of uncertainty.

Section 2 is structured as follows. In Section 2.1, we demonstrate the close relation between the variable-sized robust problem and a bicriteria optimization problem. We consider different shapes for set B in Section 2.2 and discuss the resulting variable-sized robust optimization problems. In Section 2.3, we transfer the general results of the previous section to the shortest path problem. We end the discussion of the shortest path problem with a case study.

2.1 Relation to Bicriteria Optimization

We reformulate the objective function of $\mathsf{P}(\lambda)$:

$$\max_{c\in\mathcal{U}_\lambda} c^t x = \hat{c}^t x + \max_{c\in\lambda B} c^t x = \hat{c}^t x + \max_{\tilde{c}\in B} \lambda\tilde{c}^t x = \hat{c}^t x + \lambda \max_{\tilde{c}\in B} \tilde{c}^t x = f_1(x) + \lambda f_2(x)$$

where $f_1(x) = \hat{c}^t x$ and $f_2(x) = \max_{c\in B} c^t x$. Note that both functions are convex. It is immediate that the variable-sized robust optimization problem is closely related to the bicriteria optimization problem:

$$\min_{x\in\mathcal{X}} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}$$

We define the map $F : \mathcal{X} \rightarrow \mathbb{R}^2_+, F(x) = (f_1(x), f_2(x))^t$ which maps every feasible solution in the objective space. Further, we define the polytope $\mathcal{V} = \text{conv}(\{F(x) : x \in \mathcal{X}\}) + \mathbb{R}^2_+$. We call a solution $x \in \mathcal{X}$ an *efficient extreme solution* if $F(x)$ is a vertex of $\mathcal{V}$. Denote the set of all efficient extreme solutions with $\mathcal{E}$. We call two different solutions $x \neq x'$ *equivalent* if $F(x) = F(x')$. Let $\mathcal{E}_{\min} \subset \mathcal{E}$ be a maximal subset such that no two solutions of $\mathcal{E}_{\min}$ are equivalent. The next lemma gives the direct relation between $\mathcal{E}_{\min}$ and the variable-sized robust optimization problem.

Lemma 1. *$\mathcal{E}_{\min}$ is a solution of the variable sized robust optimization problem.*

Proof. We need to prove two properties:

(I) For every $\lambda \geq 0$ there exists a solution in $\mathcal{E}_{\min}$ which is optimal for $\mathsf{P}(\lambda)$.

(II) $\mathcal{E}_{\min}$ is a smallest possible set with property (I).

(I) Let $\lambda \geq 0$ be fixed. We transfer the problem $\mathsf{P}(\lambda)$ in the objective space. The optimal value of $\mathsf{P}(\lambda)$ is equal to the optimal value of problem $\mathsf{O}(\lambda)$ since each optimal solution of this problem is a vertex of $\mathcal{V}$.

$$\min_{v \in \mathcal{V}} v_1 + \lambda v_2 \qquad (\mathsf{O}(\lambda))$$

Let v^* be the optimal solution of $\mathsf{O}(\lambda)$. By definition, $\mathcal{E}_{\min}$ contains a solution x^* with $F(x) = v^*$, i.e., an optimal solution for $\mathsf{P}(\lambda)$.
(II) Note that there is a one-to-one correspondence between vertices of $\mathcal{V}$ and solutions in $\mathcal{E}_{\min}$. Since for each vertex v' of $\mathcal{V}$ a $\lambda' \geq 0$ exists such that v' is optimal for $\mathsf{O}(\lambda)$, it follows that $\mathcal{E}_{\min}$ is indeed minimal. Note that it is important to ensure that $\mathcal{E}_{\min}$ contains no equivalent solutions. □

Different methods are known to compute $\mathcal{E}_{\min}$. The complexity of these methods depends linearly on the size of $\mathcal{E}_{\min}$. To find an additional efficient extreme solution a weighted sum problem of the form $\min_{x \in \mathcal{X}} f_1(x) + \lambda f_2(x)$ needs to be solved. In the following, we denote by T the time which is necessary to solve the nominal problem $\min_{x \in \mathcal{X}} c^T x$ and by T' the time that is necessary to solve a weighted sum problem. Note that if $f_2(x)$ is linear in x, $O(T') = O(T)$. We restate the following well known result (see e.g. [Mor86]).

Lemma 2. *$\mathcal{E}_{\min}$ can be computed in $O(|\mathcal{E}_{\min}|T')$.*

Using Lemma 1, we can transfer this result to the variable-sized robust problem.

Theorem 3. *The variable-sized robust problem can be solved in $O(|\mathcal{E}_{\min}|T')$.*

2.2 General Results

We assume that B is the unit ball of some norm. An overview of different norms and the corresponding functions $\max_{c \in \mathcal{U}_\lambda} c^t x$ is presented in Table 1. This list does not cover the large set of different uncertainty sets which is studied in the literature. For more

Name of the uncertainty	Norm $\Vert c \Vert$	Shape of B	$\max_{c\in\mathcal{U}_\lambda} c^t x$
Proportional growth	$\max_i \frac{\lvert c_i\rvert}{\hat{c}_i}$	$\times_i[-\hat{c}_i,\hat{c}_i]$	$(1+\lambda)\hat{c}^t x$
Arbitrary growth	$\max_i \frac{\lvert c_i\rvert}{d_i}, d_i > 0$	$\times_i[-d_i,d_i]$	$\hat{c}^t x + \lambda d^t x$
Constant growth	$\max_i \lvert c_i\rvert$	$\times_i[-1,1]$	$\hat{c}^t x + \lambda\Vert x\Vert_1$
Generalized Manhattan norm	$\sum_i \frac{\lvert c_i\rvert}{d_i}, d_i > 0$	$\{c : \sum_i \frac{\lvert c_i\rvert}{d_i} \leq 1\}$	$\hat{c}^t x + \lambda \max_i d_i x_i$
Generalized Euclidean norm	$\sqrt{\sum_i \frac{c_i^2}{d_i}}, d_i > 0$	$\{c : \sum_i \frac{c_i^2}{d_i} \leq 1\}$	$\hat{c}^t x + \lambda\sqrt{\sum_i d_i x_i^2}$
Ellipsoidal norm	$\sqrt{c^t Q^{-1} c}, Q \succ 0$	$\{c : c^t Q^{-1} c \leq 1\}$	$\hat{c}^t x + \lambda\sqrt{x^t Q x}$

Table 1: Overview of different shapes for the uncertainty set $\mathcal{U}_\lambda$.

information about different shapes for uncertainty sets and the resulting robust objective function we refer to [BTdHV15].

We add to the first three cases the suffix *growth* since the sets B have the form of a hyper box and, hence, each cost coefficient grows linear in λ and independent from the other coefficients. For proportional growth the coefficients c_i grow proportional to the nominal cost vector $\hat{c}$, for arbitrary growth each cost coefficient c_i has its own growth speed d_i, and for constant growth all cost coefficients grow at unit speed. For the last three cases set B has a more complex structure, we name them by the norm defining B. The generalized Euclidean norm leads to an ellipsoidal uncertainty set which is axis aligned. Whereas the more general ellipsoidal norm leads to an ellipsoidal uncertainty set with arbitrary rotation.

For proportional growth it is immediate that a nominal solution $\hat{x}$ which is optimal for cost vector $\hat{c}$ is optimal for any $\lambda \geq 0$. Hence, the optimal solution for the variable-sized robust optimization problem consists of a single element, $S = \{\hat{x}\}$.

Note that the case of constant growth leads to the concept of L_1-regularization, which is closely related to the well known Lasso method (we refer to [BBC11] for a discussion of Lasso and robust optimization).

Consider function $f_2(x) = \max_{c\in B} c^T x$ for constant growth ($f_2(x) = \|x\|_1$) and for the generalized Manhattan norm ($f_2(x) = \max_i d_i x_i$). Since we consider combinatorial optimization problems we conclude that $|\{f_2(x) : x \in \mathcal{X}\}| \leq n$ for both cases. Note that this gives an direct bound for the number of efficient extreme solutions $|\mathcal{E}_{\min}| \leq n$. We summarize these findings in the next two theorems.

Theorem 4. *The variable-sized robust optimization problem with constant growth can be solved in $O(nT)$ and $|S| \leq n$.*

Theorem 5. *The variable-sized robust optimization problem with the generalized Manhattan norm can be solved in $O(nT')$ and $|S| \leq n$.*

The next lemma helps to simplify the objective functions for the cases of the generalized Euclidean norm and the ellipsoidal norm.

Lemma 6. *Let $f_1(x), f_2(x) \geq 0 \ \forall x \in \mathcal{X}$. Each efficient extreme solution of*

$$\min_{x \in \mathcal{X}} \begin{pmatrix} f_1(x) \\ \sqrt{f_2(x)} \end{pmatrix} \tag{P_1}$$

is an efficient extreme solution of

$$\min_{x \in \mathcal{X}} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}. \tag{P_2}$$

Proof. We define the maps $cd : \mathcal{X} \to \mathbb{R}^2, cd(x) = (f_1(x), f_2(x))$ and $cd' : \mathcal{X} \to \mathbb{R}^2, cd'(x) = (f_1(x), \sqrt{f_2(x)})$. Further denote by $C_{\min} = \min_{x \in \mathcal{X}} f_1(x)$ and $D_{\min} = \min_{x \in \mathcal{X}} f_2(x)$. Let x' be an efficient extreme solution of P_1. If $cd(x')_1 = C_{\min}$ or $cd(x')_2 = D_{\min}$, it is straightforward to show that x' is also efficient extreme for P_2. Hence, we assume in the following that $cd(x')_1 > C_{\min}$ and $cd(x')_2 > D_{\min}$. We assume, for the sake of contradiction, that x' is not an efficient extreme solution of P_2. This means that two other solutions $x_1, x_2 \in \mathcal{X}$ and an $\alpha \in [0, 1]$ exists such that $\alpha cd(x_1)_1 + (1-\alpha)cd(x_2)_1 = cd(x')_1$ and $\alpha cd(x_1)_2 + (1-\alpha)cd(x_2)_2 \leq cd(x')_2$. It follows that $\alpha cd'(x_1)_1 + (1-\alpha)cd'(x_2)_1 = cd'(x')_1$ and $\alpha cd'(x_1)_2 + (1-\alpha)cd'(x_2)_2 \leq \sqrt{\alpha cd(x_1)_2 + (1-\alpha)cd(x_2)_2} \leq \sqrt{cd(x')_2} = cd'(x')_2$, since the square root is a concave and monotone function. This yields the desired contradiction, since x' is an efficient extreme solution for P_1. □

Note that instead of using the square root function on $f_2(x)$ in P_1, the proof can be extended to any increasing, concave function. Observe that $\hat{c}^t x + \lambda \sum_i d_i x_i^2 = \hat{c}^t x + \lambda d^t x$, since we consider combinatorial optimization problems. Hence, using Lemma 6 we can solve the variable-sized robust optimization problem with generalized Euclidean norm by computing the set of extreme efficient solutions for the bicriteria optimization problem with objective functions $\hat{c}^t x$ and $d^t x$. In this way may found some solutions which are not extreme efficient for the original bicriteria optimization problem with objective functions $\hat{c}^t x$ and $\sqrt{d^t x}$. After these solutions are removed, we have found the solution to the variable-sized robust optimization problem with generalized Euclidean norm.

The same approach can be applied in the case of an ellipsoidal norm. In this case, the alternative bicriteria optimization problem has the objective functions $\hat{c}^T x$ and $x^t Q x$.

2.3 Application to the Shortest Path Problem

In this section, we transfer the general results from the last section to the shortest path problem. The problem is defined on a graph $G = (V, E)$ with N nodes and M edges. We denote by $\mathcal{P}$ the set of all paths from start node $s \in V$ to end note $t \in V$. For a path $P \in \mathcal{P}$, we write $c(P) = \sum_{e \in P} c_e$ for its cost and $|P|$ for its number of edges.

2.3.1 Constant growth

We consider in more detail the implications of Theorem 4 to the shortest path problem. The corresponding bicriteria optimization problem is

$$\min_{P\in\mathcal{P}} \begin{pmatrix} \hat{c}(P) \\ |P| \end{pmatrix}$$

Note that it suffices to consider simple paths for this bicriteria optimization problem as we assume that all edge costs are positive. As each simple path contains at most N edges $|P| \in \{1,\dots,N\}$, the cardinality of $\mathcal{E}_{\min}$ is bounded by N. The computation of this set can be done at once using a labeling algorithm that stores for each path Q from s to v the cost of the path $\hat{c}(Q)$ and the number of edges contained in the path $|Q|$. Note that at each node at most N labels needs to be stored, this ensures the polynomial running time of the labeling algorithm. An alternative approach is to use an efficient procedure to compute $\mathcal{E}_{\min}$ (see [Mor86]). Each weighted sum computation corresponds to a shortest path problem.

Theorem 7. *The variable-sized robust shortest path problem for a graph $G = (V, E)$ with $|V| = N$ and $|E| = M$ with constant growth can be solved in $O(NM + N^2 \log(N))$ and $|\mathcal{S}| \leq N$.*

Proof. The nominal problem can be solved by Dijkstra's algorithm in $O(M + N\log(N))$. □

2.3.2 Generalized Manhattan norm

We consider in more detail the implications of Theorem 5 to the shortest path problem. The corresponding bicriteria optimization problem is

$$\min_{P\in\mathcal{P}} \begin{pmatrix} \hat{c}(P) \\ \max_{e\in P} d_e \end{pmatrix}$$

The set $\mathcal{E}_{\min}$ is bounded by the number of different values of d_e, $e \in E$, which is bounded by M. The computation of this set can be done at once using a labeling algorithm that stores for each path Q from s to v the cost of the path $\hat{c}(Q)$ and the most expensive edge of Q with respect to cost function d. Note that at each node at most M labels needs to be stored, this ensures the polynomial running time of the labeling algorithm. An alternative approach is to use an efficient procedure to compute $\mathcal{E}_{\min}$ (see [Mor86]). If we use Dijkstra's algorithm to solve the nominal problem we obtain the following running time.

Theorem 8. *The variable-sized robust shortest path problem for a graph $G = (V, E)$ with $|V| = N$ and $|E| = M$ with the generalized Manhattan norm can be solved in $O(M^2 + NM \log(N))$ and $|\mathcal{S}| \leq M$.*

For more details on bicriteria problems of this type, we refer to [GKR12].

2.3.3 Arbitrary Growth

Carstensen presents in her PhD thesis [Car83] bicriteria shortest path problems with two linear objective functions in which $|\mathcal{E}_{\min}| \in 2^{\Omega(\log^2(N))}$, which is not polynomial in N. Further, she proved for acylic graphs that $K \in O(n^{\log(n)})$ which is subexponential. Note that this result can directly been applied to the variable-sized robust shortest path problem with arbitrary growth.

Theorem 9. *The variable-sized robust shortest path problem on acyclic graphs with arbitrary growth can be solved in subexponential time and $|\mathcal{S}|$ is also subexponential.*

If we further restrict the graph class of G, the following results can be obtained.

Theorem 10. *Let G be a series-parallel graph with N nodes and M arcs. Then, $K \leq M-N+2$, and the variable-sized robust shortest path problem can be solved in polynomial time.*

Proof. We do a proof by an induction over the depth $D(G)$ of the decomposition tree of graph G. For the induction start, note that if $D(G) = 1$, G only consists of a single edge. Hence, there exists exactly one $s - t$ path, which is obviously also an extreme efficient solution. Therefore, $K = 1$. There are two nodes and one arc, i.e., $N = 2$ and $M = 1$. Hence, $K \leq M - N + 2$ holds.

We distinguish two cases for the induction step:

Case 1: G is a parallel composition of two series-parallel graphs G_1 and G_2.

Every path that is an extreme efficient solution for the shortest path problem in graph G is then either completely contained in G_1 or G_2, and, hence, must also be an extreme efficient solution of the shortest path instance described by G_1 or G_2. Therefore, the number of extreme efficient paths in G_1 plus the number of extreme efficient paths in G_2 is an upper bound for the number of extreme efficient paths in G. Note that $D(G_1) < D(G)$ and $D(G_2) < D(G)$. Hence, we can apply the induction hypothesis to G_1 and G_2. Denote by K_i the number of extreme efficient paths in G_i, by N_i the number of nodes, and by M_i the number of edges of G_i for $i = 1, 2$. We have that

$$\begin{aligned} K \leq K_1 + K_2 &\leq (M_1 - N_1 + 2) + (M_2 - N_2 + 2) \\ &= M_1 + M_2 - (N_1 + N_2 - 2) + 2 = M - N + 2 \end{aligned}$$

Case 2: G is a series composition of two series-parallel graphs G_1 and G_2.

Note that for every extreme efficient path, there exists an open interval $(\underline{\lambda}, \overline{\lambda})$ such that this path is the unique optimal path with respect to the weight function $\lambda c + (1-\lambda)d$ for all $\lambda \in (\underline{\lambda}, \overline{\lambda})$. Hence, we can find an ordering of all extreme efficient paths with respect to the λ values. Note that all extreme efficient paths in G must consist of extreme efficient paths of G_1 and G_2. The extreme paths of G can be obtained in the following way. We start with the shortest paths with respect to c in G_1 and G_2 and combine these two paths. Next we decrease the value of λ until either the extreme path in G_1 or G_2

changes. This change will define a new extreme path. We continue until all extreme paths of G_1 and G_2 are contained in at least one extreme path. Note that $D(G_1) < D(G)$ and $D(G_2) < D(G)$. Hence, we can apply the induction hypothesis to G_1 and G_2. Note that the number of changes is bounded by the number of extreme efficient paths minus 1. Since, for every such change, we get one additional extreme efficient path, we get in total

$$\begin{aligned} K &\leq 1 + (M_1 - N_1 + 2 - 1) + (M_2 - N_2 + 2 - 1) \\ &= M_1 + M_2 - (N_1 + N_2 - 1) + 2 \\ &= M - N + 2. \end{aligned}$$

□

Using a similar proof technique we can bound K also for layered graphs. A layered graph consists of a source node s and a destination node t and ℓ layers, each layer consists of w nodes. Node s is fully connected to the first layer, the nodes of the i^{th} layer are fully connected to the nodes of the $(i+1)^{\text{th}}$ layer, and the last layer is fully connected to node t.

Theorem 11. *Let G be a layered graph with ℓ layers and width w. Then, $K \leq (2w)^{\lceil \log(\ell+1) \rceil}$.*

Proof. Without loss of generality we can assume that $\ell = 2^k - 1$ for some $k \in \mathbb{N}$. Then, the claimed bound simplifies to $K \leq (2w)^k$. Denote by $E(t)$ the number of extreme efficient paths in G with $2^t - 1$ many layers. The base case is given by $E(0) = 1$, since the corresponding graph consists of a single edge. To bound $E(k)$ we make the following observation. Each $s-t$ path contains a single node $j \in [w]$ from the middle layer (the middle layer is the $(2^{k-1} - 1)^{\text{th}}$ layer). Hence, we can separate all $s-t$ paths into w disjoint different classes. Let E_j be the number of extreme efficient solutions in each class $j \in [w]$. The same arguments as in Case 2 in the proof of Theorem 10 can be used to show that E_j is at most the number of extreme efficient paths from s to j plus the number of extreme efficient paths from j to t minus 1. Observe further that all paths from s to a node j on the middle layer and all paths from j to t are contained in a layered graph with width w and $2^{k-1} - 1$ layers. Using these arguments we obtain the bound: $E(k) \leq w(2E(k-1) - 1)$. Using this bound we can derive the claimed bound

$$K = E(k) \leq w(2E(k-1) - 1) \leq 2wE(k-1) \leq (2w)^k E(k-k) = (2w)^k$$

□

Remark: The bound proved in Theorem 11 is subexponential for general layered graphs. But if ℓ or w is assumed to be fixed, the bound is polynomial: Since $N \approx \ell w$, $w \approx \frac{N}{\ell}$ and $\ell \approx \frac{N}{w}$, we have for ℓ fixed $K \in O\left((\frac{N}{\ell})^{\log(\ell)}\right) \in O\left(N^{\log(\ell)}\right)$ and, conversely, for w fixed $K \in O\left((2w)^{\log(\frac{N}{w})}\right) \in O\left(N^{\log(w)+1}\right)$.

Name of the uncertainty	Graph type	Computation Time
Proportional growth	General	$O(T)$
Constant growth	General	$O(NT)$
Generalized Manhattan norm	General	$O(MT)$
Arbitrary growth	Acylic	Subexponential
Arbitrary growth	Series-parallel	$O(MT)$
Arbitrary growth	Layered	$O((2w)^{\lceil \log(\ell+1) \rceil} T)$

Table 2: Results for the variable-sized robust shortest path problem.

2.3.4 Ellipsoidal Norm

The feasible set of the shortest path problem, i.e. all incidence vectors of s-t paths can be represented in the following way

$$\mathcal{X} = \left\{ x \in \{0,1\}^M : \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = b_v \ \forall v \in V \right\} \text{ with } b_v = \begin{cases} +1, & \text{if } v = s \\ -1, & \text{if } v = t \\ 0, & \text{else} \end{cases}$$

where $\delta^+(v)(\delta^-(v))$ is the set of are all edges leaving (entering) v. Note that the constraints defining $\mathcal{X}$ do not forbid cycles for the s-t paths. Nevertheless, we can use this set for the optimization problem, since all solutions containing cycles are suboptimal. The weighted sum problem which needs to be solved to compute the set of extreme efficient solutions can be represented by the following mixed integer second order cone programming problem.

$$\begin{aligned} \min\ & c^t x + r \\ \text{s.t. } & x^t Q x \le r^2 \\ & x \in \mathcal{X} \end{aligned}$$

The computational complexity of this problem is NP-complete and even APX-hard as shown in [RCH+16]. Nevertheless, real world instances can be solved exactly using modern solvers.

Before we conclude the section with a case study of the variable-sized robust shortest path problem with ellipsoidal uncertainty, we present an overview of the different results obtained in this section in Table 2.

2.3.5 Case Study

In this case study we consider the problem of finding a path through Berlin, Germany. We use a road network of Berlin which consists of $12,100$ nodes and $19,570$ edges. The data set was also used in [JMSSM05], and taken from the collection [BG16]. We use the following probabilistic model to describe the uncertain travel times of each road segment.

$$c = \hat{c} + L\xi$$

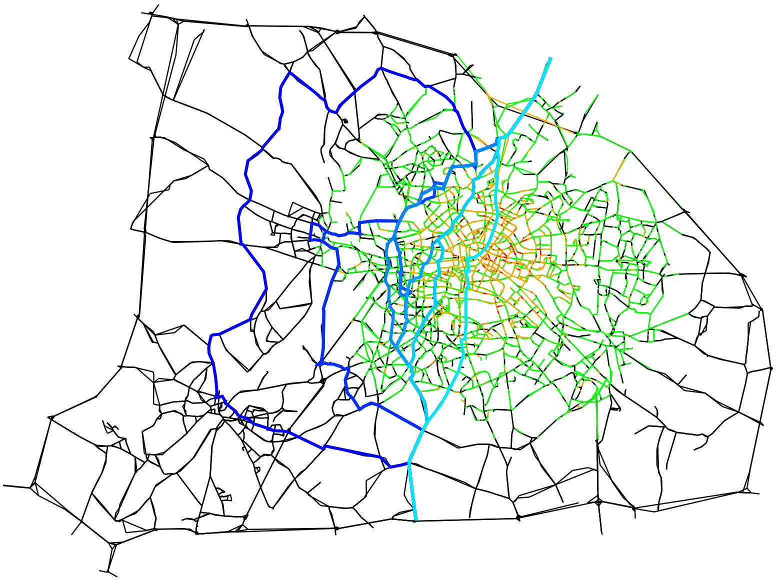

Figure 1: Berlin case study - Solution of the variable-sized robust shortest path problem. The color of each edge indicates the degree of uncertainty that affects this edge: Black - almost no uncertainty, green - small uncertainty, orange - medium uncertainty, and red - high uncertainty. The 11 different paths found as solution of the robust problem are drawn in blue: Light blue represents the nominal solution and dark blue the most robust solution.

where $\xi \sim \mathcal{N}(0, I)$ is a k-dimensional random vector which is multivariate normal distributed, $L \in \mathbb{R}^{M \times k}$, and $\hat{c}$ is taken from the data set. Under this assumption c is also multivariate normally distributed with mean $\hat{c}$ and variance LL^t. Note that the most likely realization of c form an ellipsoid. Protecting against these realizations we obtain an ellipsoidal uncertainty set.

Entries of L are chosen such that the road segments in the center of Berlin tend to be affected by more uncertainty. We compute all solutions to the variable-sized robust problem, solving the resulting mixed integer second order cone programming problem with Cplex v.12.6. Computation times were less than three minutes on a desktop computer with an Intel quad core i5-3470 processor, running at 3.20 GHz. The solution of the variable-sized robust problem contains 11 different paths. These paths are shown in Figure 1.

The start node is placed in the north of Berlin and the target node in the south. The nominal solution ignores that the center of Berlin is affected by high uncertainty and goes right through it. The most robust solution avoids all green edges, i.e., edges that are affected by small uncertainty, and takes a long detour around the center of Berlin. Beside these two extreme solutions 9 compromise solutions are found which balance uncertainty and nominal travel time.

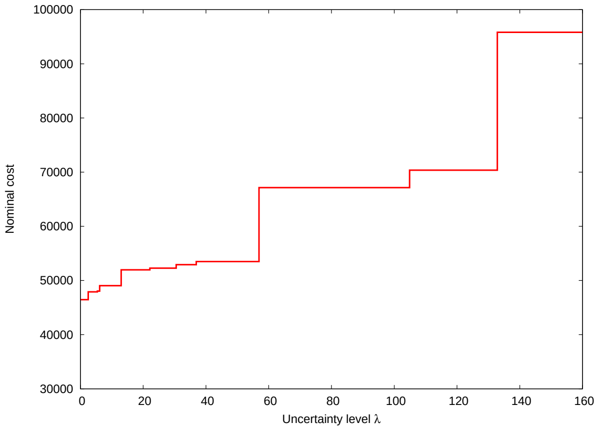

Figure 2: Berlin case study - Robustness chart. The curve indicates the nominal cost of the robust solution for some fixed uncertainty level λ.

In the robustness chart (see Figure 2) it is shown how the nominal cost ($\hat{c}^t x$) of the different compromise solution increase with an increasing level of uncertainty (λ). The chart provides the decision maker with detailed information about the cost of considering larger levels of uncertainty, and shows for how long solutions remain optimal.

We find that variable-sized robust optimization gives a more thorough assessment of an uncertain optimization problem than classic robust approaches would be able to do, while also remaining suitable with regards to computational effort.

3 Inverse Min-Max Regret Robustness

3.1 General Discussion

Having considered variable-sized approaches in the previous section, we now consider the special case of analyzing the nominal solution only.

Let $\hat{x}$ be an optimal solution to the nominal problem (P) with costs $\hat{c}$, i.e.,

$$\hat{x} \in \arg\min\{\hat{c}^t x : x \in \mathcal{X}\}$$

Note that in this case, $\hat{x}$ is also an optimizer of the min-max and min-max regret counterparts for the singleton set $\hat{\mathcal{U}} = \{\hat{c}\}$, i.e.,

$$\hat{x} \in \arg\min\left\{\max_{c \in \hat{\mathcal{U}}} c^t x : x \in \mathcal{X}\right\} \text{ and } \hat{x} \in \arg\min\left\{reg(x,\hat{\mathcal{U}}) : x \in \mathcal{X}\right\}.$$

In this section on inverse robustness, we analyze for which larger uncertainty sets these properties still hold. We focus on the min-max regret setting, as the more general variable-sized approach covers min-max counterparts in Section 2. Several ways to do so need to be differentiated.

Firstly, the "size of an uncertainty set" needs to be specified. In this section, we discuss two approaches:

- For uncertainty sets of the form

$$\mathcal{U}(\lambda) = \{c : c_i \in [(1-\lambda)\hat{c}_i, (1+\lambda)\hat{c}_i]\},$$

  the parameter $\lambda \in [0,1]$ specifies the uncertainty size, with $\mathcal{U}(0) = \hat{\mathcal{U}}$. We call these sets *regular interval sets.* Note that this corresponds to proportional growth in the previous section.

- For *general interval uncertainty sets*

$$\mathcal{U}(d^+, d^-) = \{c : c_i \in [\hat{c}_i - d_i^-, \hat{c}_i + d_i^+] \ \forall i \in [n]\},$$

  the size of $\mathcal{U}(d^+, d^-)$ is the length of intervals $|\mathcal{U}(d^+, d^-)| := \sum_{i\in[n]} d_i^- + d_i^+$, with $|\hat{\mathcal{U}}| = |\hat{\mathcal{U}}(0,0)| = 0$.

For general uncertainty sets, there are different ways to distribute a fixed amount of slack $d^- \geq 0$,$d^+ \geq 0$, all resulting in the same uncertainty size. Depending on the application, it may be useful to use upper bounds $d^- \leq M^-$ and $d^+ \leq M^+$.

Secondly, one can either look for the uncertainty set of the smallest possible size for which $\hat{x}$ is not optimal for the resulting robust objective function; or one can look for the largest possible uncertainty set for which $\hat{x}$ is still optimal in the robust sense. We refer to these approaches as worst-case or best-case inverse robustness, respectively. Our approach may also be used to choose the most robust of several candidate solutions. The problem of finding the best robust solution in a lexicographic sense has also been considered in a different setting in [IT13].

In Section 3.2, we discuss regular interval uncertainty sets, while Section 3.3 presents results on general interval sets.

3.2 Regular Interval Uncertainty Sets

3.2.1 Problem Structure

Note that $reg(x,\lambda) := reg(x,\mathcal{U}(\lambda))$ is a monotonically increasing function in λ for all x. The following calculations show that $reg(x,\lambda)$ is a piecewise-linear function in λ.

$$\begin{aligned}
reg(x,\lambda) &= \max_{c\in\mathcal{U}(\lambda)} \left(c^t x - \min_{y\in\mathcal{X}} c^t y \right) \\
&= \max_{y\in\mathcal{X}} \max_{c\in\mathcal{U}(\lambda)} c^t x - c^t y \\
&= \max_{y\in\mathcal{X}} c(x,\lambda)^t x - c(x,\lambda)^t y \\
&= \max_{y\in\mathcal{X}} (1+\lambda)c^t x - \sum_{i\in[n]} c_i(1-\lambda+2\lambda x_i) y_i
\end{aligned}$$

Here we use that the scenario $c \in \mathcal{U}(\lambda)$ maximizing $reg(x, \lambda)$ is given by $c(x, \lambda)_i = c_i(1 - \lambda + 2\lambda x_i)$. Observe that $c(x, \lambda)_i x_i = (1 + \lambda)c_i x_i$.

This gives rise to the question: Is there also any difference between worst-case and best-case inverse robustness for regular interval sets? That is, is it possible that a situation occurs as shown in Figure 3, where the regret of the nominal solution becomes larger than the regret of another solution for some λ_1, but for another $\lambda_2 > \lambda_1$, this situation is again reversed? Here, the best-case robustness approach would yield $\lambda = \lambda_2$ as the largest value of λ for which $\hat{x}$ is an optimal regret solution. However, the worst-case approach would give λ_1 as the smallest value for which $\hat{x}$ is not regret optimal. Such a situation can indeed occur, as the following example demonstrates.

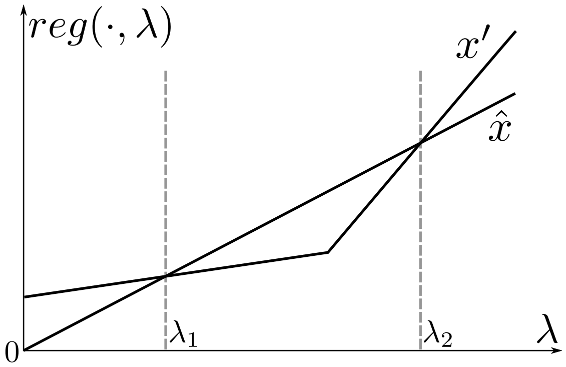

Figure 3: Comparison of the regret of two solutions $\hat{x}$ and x' for different values of λ.

Example 12. *Consider the shortest path instance presented in Figure 4. A feasible path starts at node* 1 *and ends at node* 6.

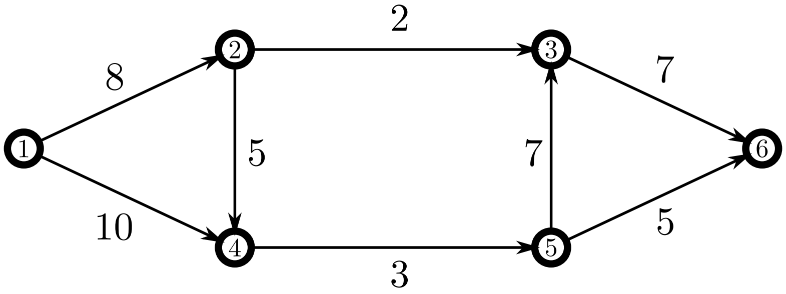

Figure 4: A sample shortest path instance with regular interval sets. The label over each edge specifies the midpoint of this edge $\hat{c}_i$.

We describe paths by the succession of nodes they visit. There are five feasible paths in the graph, $P_1 = (1, 2, 3, 6)$, $P_2 = (1, 2, 4, 5, 6)$, $P_3 = (1, 2, 4, 5, 3, 6)$, $P_4 = (1, 4, 5, 3, 6)$ *and* $P_5 = (1, 4, 5, 6)$. *The regret of each path, depending on* λ, *is shown in Table 3.*

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$reg(P_1, \lambda)$	**0.0**	2.5	6.0	9.5	13.0	**16.5**	20.0	23.5	27.0	30.5	**34.0**
$reg(P_2, \lambda)$	**4.0**	6.2	8.4	10.6	12.8	**15.0**	19.2	23.4	27.6	31.8	**36.0**
$reg(P_3, \lambda)$	13.0	16.2	20.4	24.6	28.8	33.0	37.2	41.4	45.6	49.8	54.0
$reg(P_4, \lambda)$	10.0	13.0	16.0	19.0	22.8	27.0	31.2	35.4	39.6	43.8	48.0
$reg(P_5, \lambda)$	1.0	4.5	8.0	11.5	15.0	18.5	22.0	25.5	29.0	32.5	36.0

Table 3: Regret of paths in Figure 4 depending on λ.

A unique minimizer of the nominal scenario (i.e., $\lambda = 0$*) is the path* P_1 *with a regret of zero, while path* P_2 *has a regret of four. For* $\lambda = 0.5$*, the regret of path* P_1 *becomes* 16.5*, but is only* 15 *for path* P_2*. Finally, for* $\lambda = 1$*, path* P_1 *has a regret of* 34*, but path* P_2 *has a regret of* 36*, i.e., path* P_2 *has the better regret for* $\lambda = 0.5$*, but not for* $\lambda = 0$ *and* $\lambda = 1$*. Also,* P_1 *is indeed the path with the smallest regret for* $\lambda = 1$*.*

Note how such an example is in contrast to variable-sized min-max robust optimization, where a solution that ceases to be optimal for some value λ will not be optimal again for another value $\lambda' > \lambda$. The above example can also be extended such that a path is optimal on an arbitrary number of disjoint intervals of λ.

In the following, we analyze the best-case setting in Section 3.2.2 and the worst-case setting in Section 3.2.3. We only focus on analyzing the nominal solution; an extension how this approach can be used to solve the complete variable-sized problem is given in Section 3.2.4.

3.2.2 Best-Case Inverse Robustness

The best-case inverse problem we consider here can be summarized as: Given some solution $\hat{x}$, what is the largest amount of uncertainty that can be added such that $\hat{x}$ is still optimal for the resulting min-max regret problem? The method we present in the following to answer this question can in fact be applied to any solution $x \in \mathcal{X}$, in the sense that we find the largest set $\mathcal{U}(\lambda)$ such that x is an optimal regret solution, if it exists.

More formally, the best-case inverse problem we consider here is given as:

$$\begin{aligned} \max\ & \lambda \\ \text{s.t. } & reg(\hat{x}, \lambda) \leq reg(\tilde{x}, \lambda) & & \forall \tilde{x} \in \mathcal{X} \\ & \lambda \in [0,1] \end{aligned}$$

where $reg(x, \lambda) = \max_{c \in \mathcal{U}_\lambda} \big(c^t x - opt(c) \big)$.

We now focus on combinatorial problems, where we assume that the strong duality property holds, i.e., when can solve the continuous relaxation of the problem to find an optimal solution. Example problems where this is the case include the assignment problem, the shortest path problem, or the minimum spanning tree problem. Strong duality is an important tool in classic min-max regret problems, as it allows a compact problem formulation [ABV09]. As an example, we consider the assignment problem in the following. The nominal problem is given by

$$\begin{aligned}
\min\ & \sum_{i\in[n]}\sum_{j\in[n]} \hat{c}_{ij}x_{ij} \\
\text{s.t. } & \sum_{i\in[n]} x_{ij} = 1 && \forall j \in [n] \\
& \sum_{j\in[n]} x_{ij} = 1 && \forall i \in [n] \\
& x_{ij} \in \{0,1\} && \forall i,j \in [n]
\end{aligned}$$

Using that $reg(x,\lambda) = c^t(x)x - opt(c(x))$ with

$$c_{ij}(x) = \begin{cases} (1+\lambda)\hat{c}_{ij} & \text{if } x_{ij} = 1, \\ (1-\lambda)\hat{c}_{ij} & \text{else} \end{cases}$$

and dualizing the inner optimization problem, we find a compact formulation of the min-max regret problem as follows:

$$\begin{aligned}
\min\ & \sum_{i\in[n]}\sum_{j\in[n]} (1+\lambda)\hat{c}_{ij}x_{ij} - \sum_{i\in[n]}(u_i + v_i) \\
\text{s.t. } & u_i + v_j \leq (1-\lambda)\hat{c}_{ij} + 2\lambda\hat{c}x_{ij} && \forall i,j \in [n] \\
& \sum_{i\in[n]} x_{ij} = 1 && \forall j \in [n] \\
& \sum_{j\in[n]} x_{ij} = 1 && \forall i \in [n] \\
& x_{ij} \in \{0,1\} && \forall i,j \in [n] \\
& u_i, v_i \gtrless 0 && \forall i \in [n]
\end{aligned}$$

We now re-consider the inverse problem. Note that to reformulate the constraints

$$reg(\hat{x},\lambda) \leq reg(\tilde{x},\lambda) \qquad \forall \tilde{x} \in \mathcal{X}$$

we can use the same duality approach for the left-hand side (as an optimal solution aims at having this side as small as possible), but not for the right-hand side (which should be as large as possible). Instead, for each $\tilde{x} \in \mathcal{X}$, we need to provide a primal solution. Enumerating all possible solutions in $\mathcal{X}$ as $\tilde{x}^k$, $k \in [|\mathcal{X}|]$, the inverse problem can hence be reformulated as

$$\begin{aligned}
\max\ & \lambda && (1) \\
\text{s.t. } & \sum_{i,j\in[n]} (1+\lambda)\hat{c}_{ij}\hat{x}_{ij} - \sum_{i\in[n]}(u_i + v_i)
\end{aligned}$$

$$\leq \sum_{i,j\in[n]} (1+\lambda)\hat{c}_{ij}\tilde{x}^k_{ij} - \sum_{i,j\in[n]} ((1-\lambda)\hat{c}_{ij} + 2\lambda\hat{c}_{ij}\tilde{x}^k_{ij})x^k_{ij} \qquad \forall k \in [|\mathcal{X}|] \tag{2}$$

$$u_i + v_j \leq (1-\lambda)\hat{c}_{ij} + 2\lambda\hat{c}_{ij}\hat{x}_{ij} \qquad \forall i,j \in [n] \tag{3}$$

$$\lambda \in [0,1] \tag{4}$$

$$u_i, v_i \gtreqless 0 \qquad \forall i \in [n] \tag{5}$$

$$x^k \in \mathcal{X} \qquad \forall k \in [|\mathcal{X}|] \tag{6}$$

Here, the objective function (1) is to maximize the size of the uncertainty set λ. Constraints (2) model that the regret of $\hat{x}$ needs to be at most as large as the regret of all possible alternative solutions $\tilde{x}^k$. To calculate the regret of these solutions, additional solutions x^k are required. The duality constraints in (3) ensure that the regret of $\hat{x}$ is calculated correctly.

To solve this problem, not all variables and constraints need to be included from the beginning. Instead, we can generate them during the solution process of the master problem by solving subproblems that aim at minimizing the right-hand side of Constraint (2). This is a classic min-max regret problem again.

To resolve the non-linearity between x^k and λ, we use additional variables $y^k_{ij} := \lambda x^k_{ij}$. The resulting mixed-integer program is then given as:

$$\max \lambda \tag{7}$$

$$\text{s.t.} \sum_{i,j\in[n]} (1+\lambda)\hat{c}_{ij}\hat{x}_{ij} - \sum_{i\in[n]} (u_i + v_i) \tag{8}$$

$$\leq \sum_{i,j\in[n]} (1+\lambda)\hat{c}_{ij}\tilde{x}^k_{ij} - \sum_{i,j\in[n]} (\hat{c}_{ij}x^k_{ij} + (2\tilde{x}^k_{ij} - 1)\hat{c}_{ij}y^k_{ij}) \qquad \forall k \in [|\mathcal{X}|] \tag{9}$$

$$u_i + v_j \leq (1-\lambda)\hat{c}_{ij} + 2\lambda\hat{c}_{ij}\hat{x}_{ij} \qquad \forall i,j \in [n] \tag{10}$$

$$0 \leq y^k_{ij} \leq \lambda \qquad \forall i,j \in [n], k \in [|\mathcal{X}|] \tag{11}$$

$$\lambda + x^k_{ij} - 1 \leq y^k_{ij} \leq x^k_{ij} \qquad \forall i,j \in [n], k \in [|\mathcal{X}|] \tag{12}$$

$$\lambda \in [0,1] \tag{13}$$

$$u_i, v_i \gtreqless 0 \qquad \forall i \in [n] \tag{14}$$

$$y^k_{ij} \geq 0 \qquad \forall i,j \in [n], k \in [|\mathcal{X}|] \tag{15}$$

$$x^k \in \mathcal{X} \qquad \forall k \in [|\mathcal{X}|] \tag{16}$$

For a more general formulation, which is not restricted to the assignment problem, let us assume that $\mathcal{X} = \{x : Ax \geq b, x \in \{0,1\}^n\}$ with $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$, and that strong duality holds. Then the min-max regret problem with interval sets $\mathcal{U}_\lambda$ can be formulated as

$$\min_{x\in\mathcal{X},u\in\mathcal{Y}} \big((1+\lambda)\hat{c}x - b^t u\big)$$

with $\mathcal{Y} = \{u : A^t u \leq c(x), u \geq 0\}$, see [ABV09]. Using this compact formulation for min-max regret, we can substitute Constraint (9) for

$$\sum_{i\in[n]}(1+\lambda)\hat{c}_i\hat{x}_i - b^tu \le \sum_{i\in[n]}(1+\lambda)\hat{c}_i\tilde{x}^k_i - \sum_{i\in[n]}(\hat{c}_ix^k_i + (2\tilde{x}^k_i - 1)\hat{c}_iy^k_i) \qquad \forall k\in[|\mathcal{X}|]$$

and Constraints (10) for

$$(A^tu)_i \le (1-\lambda)\hat{c}_i + 2\lambda\hat{c}_i\hat{x}_i \qquad \forall i\in[n]$$

with dual variables $u_i \ge 0$ for all $i\in[m]$.

We conclude this section by briefly considering the case where the original problem (P) does not have zero duality gap. In this case, we rewrite the constraints

$$reg(\hat{x},\lambda) \le reg(\tilde{x}^k,\lambda) \qquad \forall k\in[|\mathcal{X}|]$$

as

$$\begin{aligned}&\sum_{i\in[n]}(1+\lambda)\hat{c}_i\hat{x}_i - \sum_{i\in[n]}\left((1-\lambda)\hat{c}_i + 2\lambda\hat{c}_i\hat{x}_i\right)\bar{x}^\ell_i\\ \le&\sum_{i\in[n]}(1+\lambda)\hat{c}_i\tilde{x}^k_i - \sum_{i\in[n]}((1-\lambda)\hat{c}_i + 2\lambda\hat{c}_i\tilde{x}^k_i)x^k_i \qquad \forall k,\ell\in[|\mathcal{X}|]\end{aligned}$$

i.e., we compute the regret on both sides of the inequality by using all primal solutions as comparison. As before, variables and constraints can be generated iteratively during the solution process. To find the next solution $\bar{x}^\ell$, only a problem (P) or the original type needs to be solved.

3.2.3 Worst-Case Inverse Robustness

We now consider the worst-case inverse problem, which may be summarized as: Given some solution $\hat{x}$, what is the smallest amount of uncertainty that needs to be added such that $\hat{x}$ is not optimal for the resulting min-max regret problem anymore?

More formally, the problem we consider can be denoted as:

$$\begin{aligned}\min\ &\lambda && (17)\\ \text{s.t. } & reg(\hat{x},\lambda) \ge reg(\tilde{x},\lambda) + \varepsilon && (18)\\ & \lambda\in[0,1] && (19)\\ & \tilde{x}\in\mathcal{X} && (20)\end{aligned}$$

where ε is a small constant, i.e., we need to find an uncertainty parameter λ and an alternative solution $\tilde{x}$ such that the regret of $\tilde{x}$ is at least better by ε than the regret of $\hat{x}$.

As in the previous section, we use the assignment problem as an example how to rewrite this problem in compact form. As an optimal solution will aim at having the right-hand side of Constraint (37) as small as possible, we use strong duality to write:

$$\min\ \lambda$$

$$
\begin{aligned}
\text{s.t. } & reg(\hat{x}, \lambda) \geq \sum_{i,j \in [n]} (1+\lambda)\hat{c}_{ij}\tilde{x}_{ij} - \sum_{i \in [n]} (u_i + v_i) + \varepsilon \\
& u_i + v_j \leq (1-\lambda)\hat{c}_{ij} + 2\lambda\hat{c}_{ij}\tilde{x}_{ij} && \forall i, j \in [n] \\
& \lambda \in [0,1] \\
& \tilde{x} \in \mathcal{X}
\end{aligned}
$$

For the left-hand side of Constraint (18), we need to include an additional primal solution x' as an optimal solution for the worst-case scenario of $\hat{x}$. Linearizing the resulting products by setting $y_{ij} := \lambda x'_{ij}$ and $\beta_{ij} := \lambda \tilde{x}_{ij}$, the resulting problem formulation is then:

$$
\begin{aligned}
& \min \lambda && && (21) \\
& \text{s.t. } \sum_{i,j \in [n]} (1+\lambda)\hat{c}_{ij}\hat{x}_{ij} - \sum_{i,j \in [n]} (\hat{c}_{ij}x'_{ij} + (2\hat{x}_{ij} - 1)\hat{c}_{ij}y_{ij}) && && (22) \\
& \qquad \geq \sum_{i,j \in [n]} (\hat{c}_{ij}\tilde{x}_{ij} + \hat{c}_{ij}\beta_{ij}) - \sum_{i \in [n]} (u_i + v_i) + \varepsilon && && (23) \\
& u_i + v_j \leq (1-\lambda)\hat{c}_{ij} + 2\hat{c}_{ij}\beta_{ij} && \forall i, j \in [n] && (24) \\
& 0 \leq y_{ij} \leq \lambda && \forall i, j \in [n] && (25) \\
& \lambda + x'_{ij} - 1 \leq y_{ij} \leq \lambda && \forall i, j \in [n] && (26) \\
& 0 \leq \beta_{ij} \leq \lambda && \forall i, j \in [n] && (27) \\
& \lambda + \tilde{x}_{ij} - 1 \leq \beta_{ij} \leq \lambda && \forall i, j \in [n] && (28) \\
& \sum_{i \in [n]} x'_{ij} = 1 && \forall i \in [n] && (29) \\
& \sum_{j \in [n]} x'_{ij} = 1 && \forall j \in [n] && (30) \\
& \sum_{i \in [n]} \tilde{x}_{ij} = 1 && \forall i \in [n] && (31) \\
& \sum_{j \in [n]} \tilde{x}_{ij} = 1 && \forall j \in [n] && (32) \\
& \lambda \in [0,1] && && (33) \\
& x'_{ij}, \tilde{x}_{ij} \in \{0,1\} && \forall i, j \in [n] && (34) \\
& y_{ij}, \beta_{ij} \in [0,1] && \forall i, j \in [n] && (35)
\end{aligned}
$$

Note that this formulation has polynomially many constraints and variables, and thus can be attempted to be solved without an iterative procedure. In general, for a problem with $\mathcal{X} = \{x : Ax \geq b, x \in \{0,1\}^n\}$ and the strong duality property, we can substitute Constraint (23) for

$$
\sum_{i \in [n]} (1+\lambda)\hat{c}_i\hat{x}_i - \sum_{i \in [n]} (\hat{c}_i x'_i + (2\hat{x}_i - 1)\hat{c}_i y_i) \geq \sum_{i \in [n]} (\hat{c}_i\tilde{x}_i + \hat{c}_i\beta_i) - b^t u + \varepsilon
$$

and Constraints (24) for

$$(A^t u)_i \leq (1-\lambda)\hat{c}_i + 2\hat{c}_i\beta_i \qquad \forall i \in [n]$$

with dual variables $u_i \geq 0$ for all $i \in [m]$.

In the case that problem (P) does not have a zero duality gap, we propose to follow a similar strategy as in Section 3.2.2. That is, we consider constraints

$$\sum_{i\in[n]}(1+\lambda)\hat{c}_i\hat{x}_i - \sum_{i\in[n]}(\hat{c}_i x'_i + (2\hat{x}_i - 1)\hat{c}_i y_i)$$
$$\geq \sum_{i\in[n]}(1+\lambda)\hat{c}_i\tilde{x}_i - \sum_{i\in[n]}((1-\lambda)\hat{c}_i + 2\lambda\hat{c}_i\tilde{x}_i)x_i^k + \varepsilon \qquad \forall k \in [|\mathcal{X}|]$$

and generate them iteratively in a constraint relaxation procedure. Note that no additional variables are required.

3.2.4 Extension to the Variable-Sized Problem

Recall the relationship between the variable-sized min-max robust optimization problem and a bicriteria optimization problem discussed in Section 2, which allowed us to efficiently solve the problem. Unfortunately, such a connection cannot be derived for the variable-sized min-max regret robust optimization problem. Nevertheless, it is possible to compute a solution for the problem, i.e. a minimal set $\mathcal{S}$ such that for each $\lambda \in [0,1]$ there exists a solution $x \in \mathcal{S}$ which is optimal for the min-max regret problem with uncertainty set $\mathcal{U}(\lambda)$. In the following, we sketch a method that can be used to compute $\mathcal{S}$.

Let x be a fixed solution. Recall that $reg(x,\lambda)$ is a piecewise linear function in λ (see Figure 3). Each linear part of this function corresponds to a solution of the classic optimization problems. Hence, the function can be computed by solving a sequence of classic optimization problem. For an detailed description of the algorithm which computes the piecewise linear function $reg(x,\lambda)$ we refer to [CG16a]. A brute force method to solve the variable-sized problem is to compute for each solution $x \in \mathcal{X}$ the function $reg(x,\lambda)$. Then, we form the minimum over all these functions, i.e. we compute $F(\lambda) = \min_{x\in\mathcal{X}} reg(x,\lambda)$. Note that F is again piecewise linear. Given F, to obtain solution $\mathcal{S}$ of the variable-sized optimization problem is straightforward. Simply chose all solutions which contribute to function F. We can improve this brute force method by iteratively generating function F.

We start with some arbitrary solution x_0. We compute $reg(x_0,\lambda)$ and set $\mathcal{X}' = \{x_0\}$. Further, we define $F'(\lambda) = \min_{x\in\mathcal{X}'} reg(x,\lambda)$. The idea is to iteratively increase set $\mathcal{X}'$ until $F'(\lambda) = F(\lambda)$ for all $\lambda \in [0,1]$. To do this, we pick an interval $[\lambda_1,\lambda_2] \subset [0,1]$ for which F' is linear. Assume that $F'(\lambda) = m\lambda + b$ for all $\lambda \in [\lambda_1,\lambda_2]$. To verify that $F'(\lambda) = F(\lambda)$ for all $\lambda \in [\lambda_1,\lambda_2]$, we have to verify that for all $x \in \mathcal{X}$ it holds that

$reg(x,\lambda) \geq m\lambda + b$. This can be done by solving the following optimization problem.

$$\begin{aligned} \min\ & reg(x,\lambda) - m\lambda - b \\ & x \in \mathcal{X} \\ & \lambda \in [\lambda_1, \lambda_2] \end{aligned}$$

Denote by x' the optimal solution and by f' the optimal value of this optimization problem. Note that this problem can be transformed to a mixed integer programming problem using strong duality and the same linearization technique as used for problem (7)–(16). If $f' \geq 0$, we have verified that $F'(\lambda) = F(\lambda)$ for all $\lambda \in [\lambda_1, \lambda_2]$. If $f' < 0$, we add solution x' to $\mathcal{X}'$, compute $reg(x', \lambda)$, and update F'. In both cases, we pick a new interval for which we have not yet verified that $F'(\lambda) = F(\lambda)$. Note that this process terminates after a finite number of steps with $F' = F$.

3.3 General Interval Uncertainty Sets

3.3.1 Best-Case Inverse Robustness

We now consider consider general interval uncertainty, where the size of the uncertainty is the summed length of intervals, i.e., the best-case inverse problem we consider here is given as:

$$\begin{aligned} \max\ & \sum_{i\in[n]} d_i^+ + d_i^- \\ \text{s.t. } & reg(\hat{x}, d^+, d^-) \leq reg(\tilde{x}, d^+, d^-) && \forall \tilde{x} \in \mathcal{X} \\ & d_i^+ \in [0, M_i^+] && \forall i \in [n] \\ & d_i^- \in [0, M_i^-] && \forall i \in [n] \end{aligned}$$

where M_i^+, M_i^- denote the maximum possible deviations in each coefficient, and

$$reg(x, d^+, d^-) = \max_{c \in \mathcal{U}(d^+, d^-)} c^t x - opt(c).$$

By setting $M_i^+ = 0$ or $M_i^- = 0$ for some index i, we can model that this coefficient may not deviate in the respective direction.

We first consider this setting with the unconstrained combinatorial optimization problem, where $\mathcal{X} = \{0,1\}^n$.

Let $\hat{x}$ be optimal for $\hat{c}$. Note that for some fixed c, an optimal solution is to pack all items with negative costs. Therefore, we assume

$$\hat{x}_i = \begin{cases} 1 & \text{if } \hat{c}_i \leq 0 \\ 0 & \text{else} \end{cases}$$

There are no other optimal solutions, except for indices where $\hat{c}_i = 0$. To describe optimal solutions for the min-max regret problem under uncertainty, we make use of the following lemma:

Lemma 13. *(See [CG17]) Let* $\mathcal{U} = \times_{i\in[n]}[\hat{c}_i - d_i, \hat{c}_i + d_i]$ *for the unconstrained combinatorial optimization problem. Then, an optimal solution for* $\hat{c}$ *is also an optimal solution for the min-max regret problem.*

In our setting, this becomes:

Lemma 14. *Let* $\mathcal{U} = \mathcal{U}(d^+, d^-)$*. Then,* x^* *with*

$$x^* = \begin{cases} 1 & \textit{if } 2\hat{c}_i + d_i^+ - d_i^- \leq 0 \\ 0 & \textit{else} \end{cases}$$

is an optimal solution for the min-max regret problem.

Note that there are no other optimal solutions, except for indices where $2\hat{c}_i + d_i^+ - d_i^- = 0$. We can therefore describe the largest possible uncertainty set such that $\hat{x}$ remains optimal for the min-max regret problem in the following way:

Theorem 15. *Let an unconstrained problem with cost* $\hat{c}$ *be given. The largest uncertainty set of the form* $\mathcal{U}(d^+, d^-)$ *such that* $\hat{x}$ *remains optimal for the resulting regret problem is given by* d^+ *and* d^- *with the following properties:*

- *If* $\hat{c}_i \leq 0$*, then*

$$\begin{aligned} d_i^+ &= \min\{M_i^- - 2\hat{c}_i, M_i^+\} \\ d_i^- &= M_i^- \end{aligned}$$

- *If* $\hat{c}_i > 0$*, then*

$$\begin{aligned} d_i^+ &= M_i^+ \\ d_i^- &= \min\{M_i^+ + 2\hat{c}, M_i^-\} \end{aligned}$$

Proof. Let $\hat{c}_i \leq 0$ and $\hat{x}_i = 1$. Using Lemma 14, we choose d_i^+ and d_i^- such that $2\hat{c}_i \leq d_i^- - d_i^+$. Setting $d_i^- = M_i^-$ and solving for d_i^+, we find $d_i^+ = \min\{M_i^- - 2\hat{c}_i, M_i^+\}$. Analogously for $\hat{c}_i > 0$ and $\hat{x}_i = 0$. □

Corollary 16. *For* $M_i^- = 0$ *and* $M_i^+ = \infty$ *when* $\hat{x}_i = 1$*, and* $M_i^- = \infty$ *and* $M_i^+ = 0$ *when* $\hat{x}_i = 0$*, we have*

$$d_i^+ = -2\hat{c}_i \textit{ for } \hat{c}_i < 0 \textit{ and } d_i^- = 2\hat{c}_i \textit{ for } \hat{c}_i \geq 0$$

and the corresponding uncertainty size is hence $|\mathcal{U}(d^+, d^-)| = \sum_{i\in[n]} 2|\hat{c}_i|$.

For general combinatorial problems with the strong duality property, we can follow a similar reformulation procedure as described in Section 3.2.2. For the sake of brevity, we only give the final, linearized formulation for $\mathcal{X} = \{x : Ax \geq b, x \in \{0,1\}^n\}$ here:

$$\max \sum_{i\in[n]} d_i^+ + d_i^-$$

$$
\begin{aligned}
\text{s.t.} \sum_{i\in[n]} (\hat{c}_i + d_i^+)\hat{x}_i - b^t u & \\
\leq \sum_{i\in[n]} (\hat{c}_i + d_i^+)\tilde{x}_i^k - \sum_{i\in[n]} (\hat{c}_i x_i^k - z_i^k + \tilde{x}_i^k y_i^k + \tilde{x}_i^k z_i^k) & \qquad \forall k \in [|\mathcal{X}|] \\
(A^t u)_i \leq \hat{c}_i - d_i^- + (d_i^+ + d_i^-)\hat{x}_i & \qquad \forall i \in [n] \\
0 \leq y_i^k \leq d_i^+ & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
d_i^+ - M_i^+(1 - x_i^k) \leq y_i^k \leq M_i^+ x_i^k & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
0 \leq z_i^k \leq d_i^- & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
d_i^- - M_i^-(1 - x_i^k) \leq z_i^k \leq M_i^- x_i^k & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
Ax^k \geq b & \qquad \forall k \in [|\mathcal{X}|] \\
d_i^+ \in [0, M_i^+] & \qquad \forall i \in [n] \\
d_i^- \in [0, M_i^-] & \qquad \forall i \in [n] \\
u_i \geq 0 & \qquad \forall i \in [m] \\
y_i^k \in [0, M_i^+] & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
z_i^k \in [0, M_i^-] & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|] \\
x_i^k \in \{0, 1\} & \qquad \forall i \in [n], \forall k \in [|\mathcal{X}|]
\end{aligned}
$$

3.3.2 Worst-Case Inverse Robustness

We now consider the worst-case inverse problem with general interval uncertainty. It can be denoted as:

$$
\begin{aligned}
\min \; & \sum_{i\in[n]} d_i^+ + d_i^- && (36) \\
\text{s.t.} \; & reg(\hat{x}, d^+, d^-) \geq reg(\tilde{x}, d^+, d^-) + \varepsilon && (37) \\
& d_i^+ \in [0, M_i^+] \qquad \forall i \in [n] && (38) \\
& d_i^- \in [0, M_i^-] \qquad \forall i \in [n] && (39) \\
& \tilde{x} \in \mathcal{X} && (40)
\end{aligned}
$$

where ε is a small constant, i.e., we need to find uncertainty parameters d^+, d^- and an alternative solution $\tilde{x}$ such that the regret of $\tilde{x}$ is at least better by ε than the regret of $\hat{x}$.

Note that if $\hat{x}$ is not a unique minimizer of the nominal scenario, there is an uncertainty set $\mathcal{U}$ with arbitrary small size such that $\hat{x}$ is not the minimizer of the regret anymore. It suffices to increase an element of $\hat{x}$, which is not included in another minimizer of the nominal scenario. Hence, this approach is most relevant for unique minimizers of the nominal scenario.

As in the previous sections, we use the assignment problem as an example how to rewrite this problem in compact form. The resulting problem formulation for $\mathcal{X} = \{x :$

$Ax \geq b, x \in \{0,1\}^n\}$ is then:

$$
\begin{array}{lll}
\min & \displaystyle\sum_{i\in[n]} d_i^+ + d_i^- & \\
\text{s.t.} & \displaystyle\sum_{i\in[n]} (\hat{c}_i + d_i^+)\hat{x}_i - \sum_{i\in[n]} (\hat{c}_i x'_i - z_i + \hat{x}_i y_i + \hat{x}_i z_i) & \\
& \qquad \geq \displaystyle\sum_{i\in[n]} (\hat{c}_i \tilde{x}_i + \beta_i) - b^t u + \varepsilon & \\
& (A^t u)_i \leq \hat{c}_i - d_i^- + \beta_i + \gamma_i & \forall i \in [n] \\
& 0 \leq y_i \leq d_i^+ & \forall i \in [n] \\
& d_i^+ - M_i^+(1 - x'_i) \leq y_i \leq M_i^+ x'_i & \forall i \in [n] \\
& 0 \leq z_i \leq d_i^- & \forall i \in [n] \\
& d_i^- - M_i^-(1 - x'_i) \leq z_i \leq M_i^- x'_i & \forall i \in [n] \\
& 0 \leq \beta_i \leq d_i^+ & \forall i \in [n] \\
& d_i^+ - M_i^+(1 - \tilde{x}_i) \leq \beta_i \leq M_i^+ \tilde{x}_i & \forall i \in [n] \\
& 0 \leq \gamma_i \leq d_i^- & \forall i \in [n] \\
& d_i^- - M_i^-(1 - \tilde{x}_i) \leq \gamma_i \leq M_i^- \tilde{x}_i & \forall i \in [n] \\
& Ax' \geq b & \\
& A\tilde{x} \geq b & \\
& d_i^+, y'_i, \beta_i \in [0, M_i^+] & \forall i \in [n] \\
& d_i^-, z'_i, \gamma_i \in [0, M_i^-] & \forall i \in [n] \\
& u_i \geq 0 & \forall i \in [m] \\
& x'_i, \tilde{x}_i \in \{0,1\} & \forall i \in [n]
\end{array}
$$

3.4 Computational Insight

3.4.1 Setup

In this section we consider best-case and worst-case inverse robustness as a way to find structural insight into differences of robust optimization problems.

To this end, we used the following experimental procedure. We generated random assignment instances in complete bipartite graphs of size 15×15 (i.e., there are 225 edges). For every edge e, we generate a random nominal weight $\hat{c}_e$ uniformly in $\{0, \dots, 20\}$. We generated 2,500 instances this way.

For each instance, we solve the best-case and worst-case inverse robustness problems, where we allow symmetric deviations (i.e., $d^+ = d^-$) in the interval $[0, 20]$. Best-case problems are solved as described in Section 3.3.1 using the iterative procedure that constructs additional variables and constraints by solving a min-max regret problem as sub-procedure. Worst-case problems are solved using the compact formulation from Section 3.3.2.

Additionally, for each instance, we create 500 min-max regret problems with randomly generated symmetric interval uncertainty within the same maximum range $[0, 20]$ of possible deviations. Each min-max regret instance is solved to optimality using the compact formulation based on dualizing the inner problem. Additionally, we calculate the objective value of the nominal solution for each min-max regret instance.

To solve optimization problems, we used Cplex v.12.6 on a computer with a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 20MB cache, and Ubuntu 12.04. Processes were pinned to one core.

3.4.2 Results and Discussion

We present key values for worst-case inverse problems in Table 4. We categorized instances according to the objective value "WC" of the worst-case problem. The smallest observed objective value was 4, and the largest was 28 (the value 2 could not be achieved, as we required the difference between regret values to be at least 1).

Column "Freq" denotes how often an objective value was observed over the 2,500 instances. Columns "Reg" and "NomReg" show the average optimal regret, and the average regret of the nominal solution within each instance class, respectively. In column "Ratio", the ratio between these two values is given. Column "BC" shows the average best-case inverse value for problems within each class. The BC values are given as the negative difference to the maximum BC value, which is 9000 (i.e., smaller BC values mean that the largest possible interval uncertainty for which the nominal solution is also the optimal solution for the regret problem is smaller). "WCT" and "BCT" show the average time to solve the worst-case and the best-case inverse problems in seconds, respectively. "RegT" is the average time to solve the 500 min-max regret problems we generated per instance and is also given in seconds.

WC	Freq	Reg	NomReg	Ratio	BC	WCT	BCT	RegT
4	569	297.014	348.673	1.1739	-3.47	0.36	13.96	61.49
6	633	296.930	347.238	1.1694	-2.25	0.66	11.73	61.49
8	139	296.789	345.395	1.1637	-1.78	1.12	6.07	61.99
10	662	296.780	346.153	1.1663	-1.53	2.07	6.96	61.85
12	270	296.851	344.544	1.1606	-0.94	2.41	5.32	62.00
14	43	296.754	343.172	1.1564	-0.47	3.00	4.71	61.33
16	116	296.534	342.997	1.1567	-0.52	4.83	4.64	62.07
18	38	296.084	339.757	1.1475	-0.63	5.34	4.30	62.50
≥ 20	30	296.073	338.392	1.1429	-0.13	10.39	3.43	63.13

Table 4: Statistics for worst-case inverse problems.

Note that NomReg is decreasing with increasing WC values, i.e., when only little uncertainty is required to modify the instance such that the nominal solution is not the optimal regret solutions, then the regret of the nominal solution tends to be higher. Looking at the ratio between NomReg and Reg, we see that the quality of the nominal solution improves for larger values of WC. As the nominal solution is sometimes used as the baseline heuristic for solution algorithms (see, e.g., [CG15]), this may lead to structural insight on the performance of such algorithms for different instance classes.

The computation time for WC increases with the resulting objective value, while computation times for BC decreases for the respective instance classes. WC and BC values are connected, with instances having small WC values also tending to have large BC values. That is, if only little uncertainty is required to disturb the nominal solution, then the largest possible uncertainty set for which it is optimal also tends to be smaller. Our results do not show a significant increase in computation time for RegT, depending on WC.

Summarizing, we find that the WC value is able to categorize instances according to the relative performance of the nominal solution. While the ratio of NomReg and Reg is easier to compute than WC, if offers structural insight on why these instances behave differently, and can be used to structure benchmark sets as an example application.

4 Conclusion and Further Research

In classic robust optimization problems, one aims at finding a solution that performs well for a given uncertainty set of possible parameter outcomes. In this paper, we considered a more general problem where we assume only the shape of the uncertainty set to be given, but not its actual size.

The resulting variable-sized min-max robust optimization problem is analyzed for different uncertainty sets, and results are applied to the shortest path problem. In a brief case study, we demonstrated the value of alternative solutions to the decision maker, which can be found in little computation time.

As a special case of variable-sized uncertainty, we considered inverse problems with min-max regret objective. To solve such problems, mixed-integer programming formulations were derived. Inverse robust optimization can also be applied to give structural insight to robust optimization instances, which was demonstrated with experimental data.

Our research is the first of its kind, with possible applications in decision support, sensitivity analysis, and benchmarking.

Future research will consider inverse problems for min-max regret, and more complex uncertainty sets than hyperboxes. In particular, Γ uncertainty or ellipsoidal uncertainty might be considered. Furthermore, models and algorithms to extend our ideas to solutions which are not optimal for the nominal scenario will be helpful. One possible approach to this end is to measure the biggest difference of the regret of the given solution to the best possible regret for varying uncertainty sizes. Finally, we restricted our analysis to combinatorial problems with uncertainty in the objective; its extension to continuous problems and problems with uncertainty in the constraints is further research.

References

[ABP09] B. Alizadeh, R. E. Burkard, and U. Pferschy. Inverse 1-center location problems with edge length augmentation on trees. *Computing*, 86(4):331–343, 2009.

[ABV09] H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427 – 438, 2009.

[AO01] R. K. Ahuja and J. B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001.

[BB09] D. Bertsimas and D. B. Brown. Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6):1483–1495, 2009.

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

[BG16] H. Bar-Gera. Transportation network test problems, http://www.bgu.ac.il/∼bargera/tntp/, accessed 06/2016.

[BPS04] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510 – 516, 2004.

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

[BTdHV15] A. Ben-Tal, D. den Hertog, and J. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1):265–299, 2015.

[BTGN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, Princeton and Oxford, 2009.

[Car83] P. J. Carstensen. *The complexity of some problems in parametric linear and combinatorial programming.* University of Michigan, 1983.

[CG15] A. Chassein and M. Goerigk. A new bound for the midpoint solution in minmax regret optimization with an application to the robust shortest path problem. *European Journal of Operational Research*, 244(3):739–747, 2015.

[CG16a] A. Chassein and M. Goerigk. Compromise solutions for robust combinatorial optimization with variable-sized uncertainty. Technical report, arXiv:1610.05127, 2016.

[CG16b] A. Chassein and M. Goerigk. Performance analysis in robust optimization. In *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, pages 145–170. Springer, 2016.

[CG17] A. Chassein and M. Goerigk. Minmax regret combinatorial optimization problems with ellipsoidal uncertainty sets. *European Journal of Operational Research*, 258(1):58–69, 2017.

[CN03] E. Carrizosa and S. Nickel. Robust facility location. *Mathematical Methods of Operations Research*, 58(2):331–349, 2003.

[GKR12] J. Gorski, K. Klamroth, and S. Ruzika. Generalized multiple objective bottleneck problems. *Operations Research Letters*, 40(4):276–281, 2012.

[GS16] M. Goerigk and A. Schöbel. Algorithm engineering in robust optimization. In L. Kliemann and P. Sanders, editors, *Algorithm Engineering: Selected Results and Surveys*, volume 9220, pages 245–279. Springer, 2016.

[GYd15] B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[Heu04] C. Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, 2004.

[IT13] D. A. Iancu and N. Trichakis. Pareto efficiency in robust optimization. *Management Science*, 60(1):130–147, 2013.

[JJNS16] P. Jaillet, S. D. Jena, T. S. Ng, and M. Sim. Satisficing awakens: Models to mitigate uncertainty. Technical report, Massachusetts Institute of Technology, 2016.

[JMSSM05] O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations research*, 53(4):600–616, 2005.

[KY97] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, 1997.

[Mor86] H. I. Mordechai. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291, 1986.

[NC15] K. T. Nguyen and A. Chassein. The inverse convex ordered 1-median problem on trees under chebyshev norm and hamming distance. *European Journal of Operational Research*, 247(3):774 – 781, 2015.

[Pos13] M. Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

[RCH$^+$16] B. Rostami, A. Chassein, M. Hopf, D. Frey, C. Buchheim, F. Malucelli, and M. Goerigk. The quadratic shortest path problem: Complexity, approximability, and solution methods. *Technical Report available at www.optimization-online.org*, 2016.