

Mixed Uncertainty Sets for Robust Combinatorial Optimization

Trivikram Dokka^{*1}, Marc Goerigk^{†2}, and Rahul Roy^{‡1}

¹Department of Management Science, Lancaster University, United Kingdom

²Network and Data Science Management, University of Siegen, Germany

Abstract

In robust optimization, the uncertainty set is used to model all possible outcomes of uncertain parameters. In the classic setting, one assumes that this set is provided by the decision maker based on the data available to her. Only recently it has been recognized that the process of building useful uncertainty sets is in itself a challenging task that requires mathematical support.

In this paper, we propose an approach to go beyond the classic setting, by assuming multiple uncertainty sets to be prepared, each with a weight showing the degree of belief that the set is a "true" model of uncertainty. We consider theoretical aspects of this approach and show that it is as easy to model as the classic setting. In an extensive computational study using a shortest path problem based on real-world data, we auto-tune uncertainty sets to the available data, and show that with regard to [out-of-sample](#) performance, the combination of multiple sets can give better results than each set on its own.

Keywords: robust optimization; combinatorial optimization; uncertainty modeling; computational study

1. Introduction

In this paper we consider combinatorial problems of the form

$$\min_{\mathbf{x} \in \mathcal{X}} \mathbf{c}\mathbf{x} \tag{P}$$

with $\mathcal{X} \subseteq \{0, 1\}^n$ and uncertain cost vector \mathbf{c} . To find a solution \mathbf{x} that still performs well under the possible cost realizations, different approaches have been proposed. These include fuzzy optimization [KZ10], stochastic programming [BL11], or robust optimization [GS16, GMT14].

*Email: t.dokka@lancaster.ac.uk

†Email: marc.goerigk@uni-siegen.de

‡Corresponding author. Email: r.roy@lancaster.ac.uk

In the robust optimization approach, we assume that all possible cost realizations \mathbf{c} are modelled through a so-called uncertainty set \mathcal{U} , and we want to protect against all outcomes without knowledge of a probability distribution. The resulting worst-case problem is then of the form

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

Special cases of this kind have been investigated thoroughly (see [KZ16, BK18] for recent overviews on robust combinatorial optimization). A common assumption in these settings is that the shape of the uncertainty set \mathcal{U} is known, e.g., one commonly assumes that the uncertainty is

$$\begin{aligned} \text{interval-based } \mathcal{U} &= \prod_{i \in [n]} [\underline{c}_i, \bar{c}_i], \\ \text{discrete } \mathcal{U} &= \{\mathbf{c}^1, \dots, \mathbf{c}^K\}, \\ \text{ellipsoidal } \mathcal{U} &= \{\mathbf{c} \in \mathbb{R}_+^n : (\mathbf{c} - \hat{\mathbf{c}})\mathbf{\Sigma}^{-1}(\mathbf{c} - \hat{\mathbf{c}}) \leq \lambda\}, \\ \text{budgeted } \mathcal{U} &= \{\mathbf{c} : c_i = \underline{c}_i + z_i \bar{c}_i, \sum_{i \in [n]} z_i \leq \Gamma, z_i \in \{0, 1\}\}, \\ \text{or polyhedral } \mathcal{U} &= \{\mathbf{c} : V\mathbf{c} \leq \mathbf{d}, \mathbf{c} \geq \mathbf{0}\}. \end{aligned}$$

Note that in the min-max setting, using a discrete uncertainty set is equivalent to using its convex hull $\mathcal{U} = \text{conv}(\{\mathbf{c}^1, \dots, \mathbf{c}^K\})$. So far, comparatively little research has investigated how a decision maker can actually come up with an uncertainty set that produces a robust solution in accordance with his or her wishes. In the data-driven approach [BGK18], we do not assume an uncertainty set to be given, but only data observations, which are usually discrete. These are then used to construct uncertainty sets, e.g., using different approaches from statistical testing.

In [CG18a, CG18c], the authors considered a setting in which the shape of the uncertainty set is given, but not its size. Models are introduced by which compromise robust solutions can be found, which perform well on average over all considered uncertainty sizes. Furthermore, in [CKR18] the sensitivity of robust solutions to the uncertainty size was considered.

In the recent paper [CDG19], real-world data modeling traffic speeds in the city of Chicago were used to test different uncertainty sets for shortest path problems experimentally. In particular, discrete uncertainty and ellipsoidal uncertainty sets were found to produce a good trade-off on an out-of-sample evaluation set of scenarios with respect to average performance, worst-case performance and the conditional value at risk (CVaR) criterion.

In this paper, we introduce a novel approach to handle uncertainty in robust optimization. This mixed-uncertainty setting is a direct generalization of the classic robust optimization approach, where we protect against multiple uncertainty sets simultaneously. We demonstrate that this approach is well-suited to data-driven settings, where the decision maker is not able to determine the shape and size of uncertainty a priori. By using *irace* [LIDLSB11], a software designed to tune algorithms automatically, we determine the best-performing combination of uncertainty sets for the same Chicago test data as used in [CDG19]. Our proposed approach is generally applicable, and possible to implement using off-the-shelf software with little theoretical knowledge from the decision maker, making it flexible and attractive for practical purposes.

The rest of this paper is structured as follows. In Section 2, we formally introduce our setting of mixed uncertainty sets for uncertain combinatorial problems. We demonstrate how to find compact mixed-integer programming models, and point out cases solvable in polynomial time. Section 3 presents an extensive computational case study, highlighting the usefulness of our approach for a real-world shortest path problem. We conclude this paper in Section 4 and point out further research questions.

2. Mixed Uncertainty Sets

In classic min-max robust combinatorial optimization, we consider the problem

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{c} \in \mathcal{U}} \mathbf{c}\mathbf{x} \quad (\text{RP})$$

where the uncertainty set \mathcal{U} contains all possible outcomes of the cost vector \mathbf{c} . As a direct generalization, it is possible that multiple uncertainty sets $\mathcal{U}_1, \dots, \mathcal{U}_N$ need to be considered. For each $j \in [N]$, we are given a weight w_j that denotes the importance to protect against \mathcal{U}_j , or likelihood of its occurrence.

The resulting weighted robust optimization is then as follows:

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{j \in [N]} w_j \max_{\mathbf{c} \in \mathcal{U}_j} \mathbf{c}\mathbf{x} \quad (\text{WRP})$$

There are different settings where the application of a model of the form (WRP) can be useful. For example: (1) We receive different forecasts for future developments of costs, each providing a set of most likely scenarios. Instead of using only one of these sets or merging them, we consider a weighted robust problem, where we assign each forecast an expert estimate w_j whether it can be trusted. (2) We create different uncertainty sets based on different degrees of risk-willingness. We then find a single compromise solution against all levels of risk. For example, one uncertainty set may cover the worst 80% of outcomes of a multivariate normal distribution, whereas another uncertainty set covers 95%. We can find a single solution hedging against both worst cases with prescribed weights. (3) We have an original set of observations for our uncertain data, and do not know which shape of uncertainty set may be appropriate for the problem at hand. Selecting the (right) uncertainty set is itself a decision problem under uncertainty. So far, researchers left this problem to the user/decision-maker and developed approaches for robust optimization after this decision is made offering different choices to this decision. However, this itself can be posed as an optimization or a learning problem. In other words, ideally, we would want to learn the right uncertainty set from data, which may be a mix of many known sets. Also, automatically learning a mixed uncertainty set allows the possibility for the set to be dynamic which can change with a change in the random nature of the underlying uncertain parameters.

Mixing uncertainty sets and tuning the mixing (hyper) parameters to learn the best mix is a challenging task. Research in this direction is in its early stages, the only work to have experimented in this direction is that of [CH15] where authors consider combining ellipsoidal sets within a Bayesian setting for robust linear programs. Moreover, even under the assumption that we can formulate this learning task and embed it in a holistic algorithmic framework for data-driven robust

optimization, there is no guarantee that mixing sets will help in improved robust solutions for any choice of robustness measures. In this spirit, (WRP) is a first step towards automating selection of the right uncertainty set. In this work we would like to investigate if mixing (known) sets can give better robust solutions.

We now consider the complexity of (WRP), and possibilities to formulate it using mixed-integer programming. Note that (WRP) can be written as a compact optimization problem if that is possible for each \mathcal{U}_i , by combining each model. As an example, if \mathcal{U}_1 is a budgeted uncertainty set of the form $\mathcal{U}_1 = \{\mathbf{c} : c_i = \underline{c}_i + z_i \bar{c}_i, \sum_{i \in [n]} z_i \leq \Gamma, z_i \in \{0, 1\}\}$ (see [BS03]), then the robust problem (RP) can be written as

$$\begin{aligned} \min \quad & \sum_{i \in [n]} \underline{c}_i x_i + \Gamma \pi + \sum_{i \in [n]} \rho_i \\ \text{s.t.} \quad & \pi + \rho_i \geq (\bar{c}_i - \underline{c}_i) x_i \quad \forall i \in [n] \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

If \mathcal{U}_2 is a polyhedral uncertainty set of the form $\mathcal{U}_2 = \{\mathbf{c} : V\mathbf{c} \leq \mathbf{d}, \mathbf{c} \geq \mathbf{0}\}$, then its robust problem (RP) can be written as

$$\begin{aligned} \min \quad & \mathbf{d}\boldsymbol{\alpha} \\ \text{s.t.} \quad & V^t \boldsymbol{\alpha} \geq \mathbf{x} \\ & \boldsymbol{\alpha} \geq \mathbf{0} \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

Combining both uncertainty sets in a weighted robust problem (WRP) using weights w_1 for budgeted uncertainty and w_2 for polyhedral uncertainty, the combination of models yields

$$\begin{aligned} \min \quad & w_1 \left(\sum_{i \in [n]} \underline{c}_i x_i + \Gamma \pi + \sum_{i \in [n]} \rho_i \right) + w_2 \mathbf{d}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \pi + \rho_i \geq (\bar{c}_i - \underline{c}_i) x_i \quad \forall i \in [n] \\ & V^t \boldsymbol{\alpha} \geq \mathbf{x} \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

As (WRP) is an generalization of (RP), it is at least as hard. Hence, cases of (WRP) that involve uncertainty sets \mathcal{U}_i for which we already know that the classic robust optimization problem is NP-hard, will be NP-hard as well.

We therefore focus on cases where (RP) is still polynomially solvable. First, notice that when $\mathcal{U}_1, \dots, \mathcal{U}_N$ are interval uncertainty sets of the form $\mathcal{U}_j = \prod_{i \in [n]} [\underline{c}_i^j, \bar{c}_i^j]$, then

$$\sum_{j \in [N]} w_j \max_{\mathbf{c} \in \mathcal{U}_j} \mathbf{c}\mathbf{x} = \sum_{i \in [n]} \left(\sum_{j \in [N]} w_j \bar{c}_i^j \right) x_i$$

which is a problem of nominal type. Hence, (WRP) with interval sets has the same complexity as the nominal problem (P).

Now let $\mathcal{U}_1, \dots, \mathcal{U}_N$ be budgeted uncertainty sets with $\mathcal{U}_j = \{\mathbf{c} : c_i = \underline{c}_i^j + z_i \bar{c}_i^j, \sum_{i \in [n]} z_i \leq \Gamma^j, \mathbf{z} \in \{0, 1\}^n\}$. We can write (WRP) as

$$\min \sum_{j \in [N]} w_j \left(\sum_{i \in [n]} \underline{c}_i^j x_i + \Gamma^j \pi^j + \sum_{i \in [n]} \rho_i^j \right) \quad (\text{B-WRP})$$

$$\begin{aligned} \text{s.t. } \pi^j + \rho_i^j &\geq (\bar{c}_i^j - \underline{c}_i^j)x_i && \forall i \in [n], j \in [N] \\ \mathbf{x} &\in \mathcal{X} \end{aligned}$$

Lemma 1. *There is an optimal solution to (B-WRP), where for all $j \in [N]$, we have $\pi^j = \bar{c}_{i(j)}^j - \underline{c}_{i(j)}^j$ for some $i(j) \in [n]$, or $\pi^j = 0$.*

Proof. For some fixed \mathbf{x} , the remaining optimization problem in π^j and ρ^j can be decomposed into N independent sub-problems. For each sub-problem, the result is known from the classic budgeted uncertainty case (see [BS03]). \square

Theorem 2. *For constant N , (B-WRP) can be solved in polynomial time, if (P) can be solved in polynomial time.*

Proof. We rewrite (B-WRP) as

$$\begin{aligned} \min \sum_{j \in [N]} w_j &\left(\sum_{i \in [n]} \underline{c}_i^j x_i + \Gamma^j \pi^j + \sum_{i \in [n]} \left[(\bar{c}_i^j - \underline{c}_i^j)x_i - \pi^j \right]_+ \right) \\ \text{s.t. } \mathbf{x} &\in \mathcal{X} \end{aligned}$$

where $[y]_+ = \max\{y, 0\}$. Note that

$$\left[(\bar{c}_i^j - \underline{c}_i^j)x_i - \pi^j \right]_+ = \left[\bar{c}_i^j - \underline{c}_i^j - \pi^j \right]_+ x_i$$

as $\pi \geq 0$. Using Lemma 1, we enumerate all $(n+1)^N$ options for the π^j variables. For fixed values of $\boldsymbol{\pi}$, (B-WRP) reduces to

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{i \in [n]} \left(\sum_{j \in [N]} w_j \left(\underline{c}_i^j + \left[\bar{c}_i^j - \underline{c}_i^j - \pi^j \right]_+ \right) \right) x_i + \sum_{j \in [N]} w_j \Gamma^j \pi^j,$$

which is of the nominal type. \square

Note that this approach cannot be used if N is unbounded (i.e., part of the input). For this case, we can still identify cases that are tractable in the following.

For $\mathbf{x} \in \mathcal{X}$, denote by $X = \{i \in [n] : x_i = 1\}$ the corresponding set of chosen items. We rewrite the objective of (WRP) as

$$f(X) = \sum_{j \in [N]} w_j \left(\sum_{i \in X} \underline{c}_i^j + D^j(X) \right)$$

where $D^j(X)$ denotes the sum of the Γ^j largest values $d_i^j := \bar{c}_i^j - \underline{c}_i^j$ for $i \in X$, i.e.,

$$D^j(X) = \max \left\{ \sum_{i \in X} d_i^j z_i : \sum_{i \in [n]} z_i \leq \Gamma^j, z \in \{0, 1\}^n \right\}$$

Theorem 3. *Function f is submodular.*

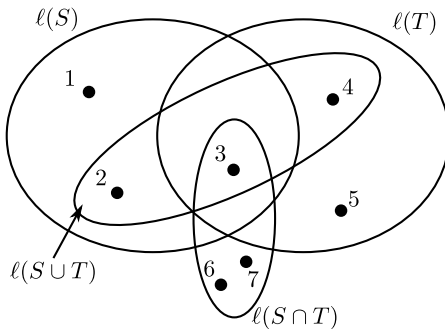


Figure 1: Illustration for the proof of Theorem 3.

Proof. We first show that D^j is a submodular function. As the sum of submodular functions is still submodular, it follows that f is submodular.

To see that D^j is submodular, let $S, T \subseteq [n]$ be given. Let $\ell(X)$ be the up to Γ^j items that maximize $\sum_{i \in \ell(X)} d_i^j$. Note that $\ell(S \cup T) \subseteq \ell(S) \cup \ell(T)$, while $\ell(S \cap T)$ is not necessarily in $\ell(S) \cup \ell(T)$. Figure 3 illustrates these relationships.

For any element $i \in \ell(S \cap T)$, one of two cases hold: If $i \in \ell(S \cup T)$, then also $i \in \ell(S)$ and $i \in \ell(T)$. If $i \notin \ell(S \cup T)$, then there is an element $k \in (\ell(S) \cup \ell(T)) \setminus \ell(S \cup T)$ with $d_k^j \geq d_i^j$. Hence,

$$D^j(S) + D^j(T) \geq D^j(S \cup T) + D^j(S \cap T)$$

and the claim follows. \square

Recall that a matroid consists of a finite set S and a collection \mathcal{S} of subsets of S , such that (i) $\emptyset \in \mathcal{S}$, (ii) if $A \in \mathcal{S}$ and $B \subseteq A$, then $B \in \mathcal{S}$, and (iii) if $A, B \in \mathcal{S}$ with $|A| = |B| + 1$, there exists $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{S}$ (see, e.g., [Wel10]). As it is possible to optimize a submodular function efficiently over a matroid, we reach the following conclusion.

Corollary 4. *For problems where \mathcal{X} is a matroid, (B-WRP) can be solved in polynomial time, even for unbounded N .*

Corollary 4 applies to, e.g., the weighted robust spanning tree problem, or the weighted robust selection problem. These results also apply to combinations of interval and budgeted uncertainty sets.

Note that the above reasoning can also be extended to (WRP) using knapsack uncertainty, see [Pos18]. In this setting, each \mathcal{U}_j is given as a multi-knapsack polytope with $|S|$ knapsack constraints, i.e.,

$$\mathcal{U}_j = \left\{ \mathbf{c} : \sum_{i \in [n]} a_{ki}^j c_i \leq b_k, k \in S, \mathbf{0} \leq \mathbf{c} \leq \bar{\mathbf{c}}^j \right\}$$

If $|S| = 1$, then the worst-case problem can be solved by sorting all items in X with respect to a_{ki}^j/b_k . Recall that a function f is submodular, if and only if $f_u(A) \geq f_u(B)$ for all $A \subseteq B$ and $u \in [n] \setminus B$, where $f_u(S) = f(S \cup \{u\}) - f(S)$ is the marginal contribution of u . As the set A will sort its items in the same order as

they appear in B , the marginal contribution of an added element will be at least as large to A as it is to B .

This sorting argument does not hold in the case of $|S| \geq 2$. As an example, consider

$$\mathcal{U} = \{\mathbf{c} \in [0, 1]^3 : c_1 + c_2 \leq 1, c_2 + c_3 \leq 1\}$$

For $S = \{1, 2\}$ and $T = \{2, 3\}$, we have $f(S) = 1$, $f(T) = 1$, $f(S \cup T) = 2$ and $f(S \cap T) = 1$ (modifying the definition of f accordingly).

Finally, we consider the approximability of problems with discrete uncertainty sets. It is a well-known result (see, e.g., [ABV09, CG18b]) that for classic robust combinatorial optimization problems with discrete uncertainty $\mathcal{U} = \{\mathbf{c}^1, \dots, \mathbf{c}^K\}$, an optimal solution to the midpoint scenario $\frac{1}{K} \sum_{k \in [K]} \mathbf{c}^k$ gives a K -approximation. We extend this result to mixed discrete uncertainty sets.

Let $\mathcal{U}^j = \{\mathbf{c}^{1,j}, \dots, \mathbf{c}^{K_j,j}\}$ for all $j \in [N]$. We construct the mixed-uncertainty midpoint scenario

$$\hat{\mathbf{c}} = \sum_{j \in [N]} w_j \left(\frac{1}{K_j} \sum_{k \in [K_j]} \mathbf{c}^{k,j} \right)$$

and set $K^{\max} = \max_{j \in [N]} K_j$.

Theorem 5. *An optimal solution to problem (P) with costs $\hat{\mathbf{c}}$ is a K^{\max} -approximation to (WRP) with mixed discrete scenarios.*

Proof. Let $\hat{\mathbf{x}}$ be an optimal solution to $\hat{\mathbf{c}}$, and let \mathbf{x}^* be an optimal solution to (WRP). Then

$$\begin{aligned} \sum_{j \in [N]} w_j \max_{k \in [K_j]} \mathbf{c}^{k,j} \hat{\mathbf{x}} &\leq \sum_{j \in [N]} w_j \left(\sum_{k \in [K_j]} \mathbf{c}^{k,j} \hat{\mathbf{x}} \right) \\ &\leq K^{\max} \sum_{j \in [N]} w_j \left(\frac{1}{K_j} \sum_{k \in [K_j]} \mathbf{c}^{k,j} \hat{\mathbf{x}} \right) \\ &\leq K^{\max} \sum_{j \in [N]} w_j \left(\frac{1}{K_j} \sum_{k \in [K_j]} \mathbf{c}^{k,j} \mathbf{x}^* \right) \\ &\leq K^{\max} \sum_{j \in [N]} w_j \max_{k \in [K_j]} \mathbf{c}^{k,j} \mathbf{x}^* \end{aligned}$$

Hence, the objective of solution $\hat{\mathbf{x}}$ in problem (WRP) is at most K^{\max} times the objective of an optimal solution. \square

While Theorem 5 holds for any N , we can get stronger approximation results for bounded values of N , as the following result demonstrates.

Theorem 6. *If there exists an $\alpha(K^{\max})$ -approximation to the min-max problem with K^{\max} scenarios, then there exists an $\alpha((K^{\max})^N)$ -approximation to (WRP). In particular, for constant values of K^{\max} and N , there exists an FPTAS for (WRP) if the underlying problem (P) is shortest path, minimum spanning tree, selection, representative selection, or min-knapsack.*

Proof. Let discrete uncertainty sets $\mathcal{U}^j = \{\mathbf{c}^{1,j}, \dots, \mathbf{c}^{K_j,j}\}$ for all $j \in [N]$ be given. We construct the following set, consisting of all combinations of scenarios:

$$\mathcal{U}' = \left\{ \sum_{j \in [N]} w_j \mathbf{c}^{k_j,j} : k_j \in [K_j] \text{ for all } j \in [N] \right\}$$

Note that $|\mathcal{U}'| \leq (K^{max})^N$ and

$$\sum_{j \in [N]} w_j \max_{k \in [K_j]} \mathbf{c}^{k,j} \mathbf{x} = \max_{k_1, \dots, k_N} \sum_{j \in [N]} w_j \mathbf{c}^{k_j,j} \mathbf{x} = \max_{\mathbf{c} \in \mathcal{U}'} \mathbf{c} \mathbf{x}$$

Hence, (WRP) is equivalent to a min-max optimization problem with at most $(K^{max})^N$ scenarios. The existence of an FPTAS for specific problems of type (WRP) thus follows from the existence of an FPTAS for the min-max problem, see, e.g., [KZ16]. \square

3. Tuning Experiments

Algorithms which solve optimization problems involve a number of parameters, and these parameters can be carefully tuned so that the performance of the algorithms is optimized. For this purpose, a tuning tool to automatically configure optimization algorithms called *irace* [LIDLSB11] has been developed. We use *irace* to find the best-performing combination of uncertainty sets for (WRP) a given set of instances of the robust shortest path problem.

3.1. Experimental Setup

We used the same real-world shortest path data as first introduced in [CDG19]. The graph models the City of Chicago, and consists of 538 nodes and 1308 arcs. The *morning* data set was used to model the problem uncertainty, containing 271 scenarios that represent morning rush hours during week days. Each scenario contains the travel speed for each arc.

In our experiments, three uncertainty sets, ellipsoidal, convex hull and interval, are used in generating the mixed uncertainty sets. Additionally, we set $N = 3$, meaning that we combine up to three uncertainty sets; however, the tuning algorithm can also choose less. We refer to these three sets as the *parent* uncertainty sets. The choice of parent uncertainty sets is driven by computational efficiency and relative performance of uncertainty sets demonstrated in [CDG19]. It is possible that two parent sets are of the same type, e.g., a combination that uses two convex hull uncertainty sets and one ellipsoidal uncertainty set. Corresponding to each parent set, we have a scaling parameter λ , which takes up values in pre-defined intervals (including the interval limits). The intervals are so chosen to reflect a reasonable range of choices for a decision maker. For ellipsoidal uncertainty we use $\lambda \in [0, 20]$, while we use $\lambda \in [0, 1]$ for interval and convex hull uncertainty. Parameter λ represents the size of the uncertainty set, in relation to the observed scenarios. For a formal parameter definition, we refer to [CDG19].

Moreover, we also associate a weight w_j to each parent set which corresponds to the significance of the parent in the mixed uncertainty set and, therefore, in the objective function.

In our experiments, our objective is to find a path that is robust when driving during morning rush hours, modeled through 271 scenarios. We use 75% of the scenarios (in-sample data) to construct the uncertainty sets, and we evaluate solutions in-sample and out-of-sample separately. To this end, we generate 600 random $s-t$ pairs over the node set, which fulfill a minimum distance criterion to avoid nodes that are close to each other. To find a balanced evaluation of all methods, we use three performance criteria: (1) The average objective function value over all $s-t$ pairs and all scenarios, denoted as Avg. (2) The average of the worst-case objective function value for each $s-t$ pair, denoted as Max. (3) The average value of the worst 5% of objective values for each $s-t$ pair (as in the conditional value at risk measure), denoted as CVaR. We see that each evaluation is comprised of three performance measures, i.e., each solution is assigned a three-dimensional objective vector.

For *irace* to tune the algorithmic parameters of mixed uncertainty sets, the algorithm must return the cost function as a single value. Hence, three additional parameters were defined, with each parameter being a weight to each performance measure in the objective vector. The cost function then becomes a weighted sum of the performance measures. However, it is important to note that these three weights are user-defined and not automatically configured by *irace*. The three weights corresponding to their performance measures are sampled from the interval $[0, 1]$ with a step size of 0.1, and only those values are retained that sum up to one (which makes a total of 66 such combinations of weights). Each combination results in an algorithm, and therefore, a solution for the shortest path problem using the mixed uncertainty sets. Hence, in total, for mixed uncertainty sets, we have nine automatically configurable parameters and three fixed user-defined parameters. We used a fixed computational budget of 10,000 experiments for a given *irace* run.

As a comparison, we generate 41 possible values for the scaling parameter λ when each parent uncertainty set is separately used to compute solutions to the robust shortest path problem; this does not involve any parameter tuning as solutions are computed for different sizes of the parent uncertainty sets. The 41 values are equidistant, i.e., for ellipsoidal sets we use a step size of 0.5 for λ , whereas for interval and convex hull we use a step size of 0.025. We use the 41 models obtained for each parent set to compare the performance of mixed set solutions with the performance of the solutions delivered by the pure parent sets.

3.2. Results

We present the performance of the uncertainty sets, both mixed and parent, in Figures 2 and 3. As mentioned earlier, a total of 41 objective space vectors are obtained for each parent uncertainty set, [whereas](#) a total of 66 objective space vectors are obtained for the mixed uncertainty sets. Each objective space vector among the 66 vectors for mixed uncertainty sets corresponds to a unique best parametric configuration obtained for different combinations of the weights of the performance measures. Moreover, each element (performance measure) of the objective space vector corresponding to a unique configuration is an average of all the values of that performance measure taken over all the scenarios and $s-t$ pairs. This also holds true for the solutions obtained using the parent uncertainty sets. Figure 2 shows the trade-off curves between the Avg and the Max performance measures for both in-sample and out-of-sample data, while Figure 3 shows the trade-off curves between the Avg and the CVaR performance measures for both in-sample and out-

of-sample data. All the values are in unit of minutes; a lower value indicates better performance. Hence, good trade-off solutions should move from top left to the bottom right of the curves. Some values for interval sets are not visible, as they are outside the plotted ranges.

Figure 2(a) shows that for in-sample performance, convex hull solutions dominate all other solutions for the Avg vs Max trade-offs by construction; however, the mixed set solutions closely match the convex hull solutions. Interval solutions perform the worst among all, especially for higher values of the scaling parameter. Figure 3(a) shows that for in-sample performance, ellipsoidal solutions exhibit the best Avg vs CVaR trade-offs among all the solutions but are closely matched by the mixed set solutions. However, convex hull solutions perform worse than both ellipsoidal and mixed set, and interval solutions closely match the performance of the convex hull solutions. Summarily, both ellipsoidal and convex hull solutions do not exhibit stability across both the trade-off curves, i.e., while convex hull dominates the in-sample Avg vs Max curves, ellipsoidal exhibits best performance for in-sample Avg vs CVaR trade-offs. Mixed uncertainty set solutions exhibit stability across both in-sample Avg vs Max and in-sample Avg vs CVaR trade-offs as they closely match the respective performance of both convex hull and ellipsoidal solutions.

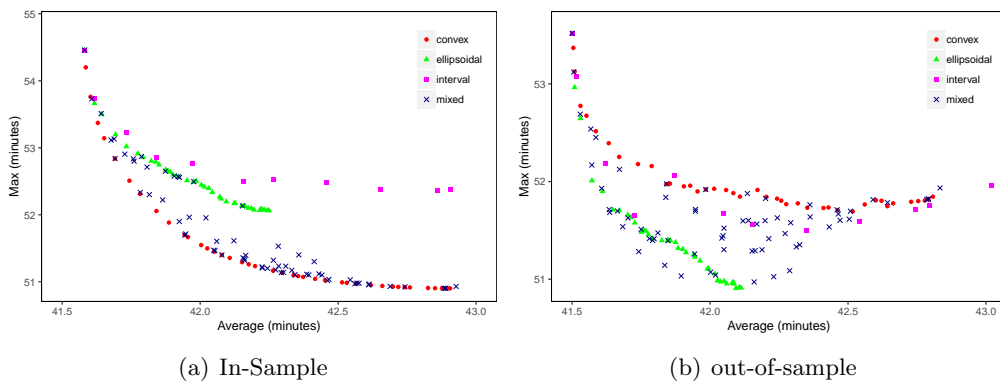


Figure 2: Average vs Worst-Case Performance

When out-of-sample performance is considered for both Avg vs Max and Avg vs CVaR trade-offs (Figures 2(b) and 3(b)), mixed set solutions can dominate all other solutions. Convex hull solutions perform even worse than interval solutions for both the trade-offs, implying that they are over-fitted to the data. The performance of ellipsoidal solutions improve from in-sample to out-of-sample compared to the performance of convex hull and interval solutions for Avg vs Max trade-offs, but lose their best trade-off performance for Avg vs CVaR only to mixed set solutions.

The key aspect to note is that, even though mixed solutions only use the three parent sets, their combination can outperform each separately, i.e., we can observe a synergy effect when mixing uncertainty sets.

While the performance comparison among all the uncertainty sets help us establish that mixed uncertainty sets not only exhibit stability over both in-sample and out-of-sample data but also perform better than the parent sets for certain configurations, we do not observe how each uncertainty set performs for each s - t pair separately, as

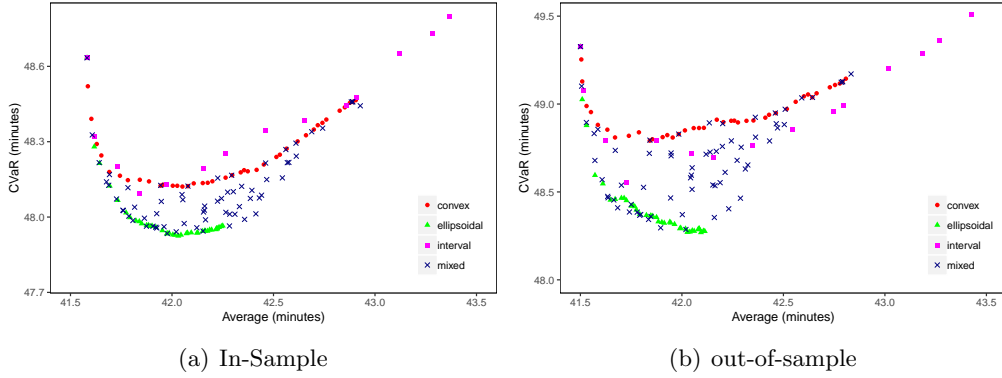


Figure 3: Average vs CVaR Performance

we explained earlier that each solution delivered by an uncertainty set is an average of all the values taken over all the scenarios and s - t pairs. In Appendix A, we present additional insight on representative configurations.

We summarize our findings. Convex hull solutions show good in-sample performance for Avg vs Max trade-offs, but are not stable when facing out-of-sample scenarios. The in-sample performance of convex hull solutions is closely matched by the mixed set solutions. Convex hull solutions perform worse for Avg vs CVaR trade-offs across both in-sample and out-of-sample scenarios.

Interval solutions do not perform well in general, but are easy to find and can be a reasonable approach for smaller values of the scaling parameter.

Ellipsoidal solutions exhibit stability over both in-sample and out-of-sample performance for both Avg vs Max and Avg vs CVaR trade-offs and offer a good approach over a wide range of the scaling parameter. In addition, they also deliver the best CVaR performance across both in-sample and out-of-sample data among all the uncertainty sets. However, solutions given by ellipsoidal set are dominated by the mixed set solutions in the region of best trade-offs for both in-sample and out-of-sample scenarios when averaged over all the s - t pairs and scenarios.

Mixed set solutions closely match the in-sample performance of convex hull and ellipsoidal solutions for Avg vs Max and Avg vs CVaR trade-offs respectively, but they dominate all the other solutions when facing out-of-sample scenarios, especially in the region of best trade-offs, i.e., mixed set solutions are found to deliver the best solutions among all the uncertainty sets for certain configurations. Besides, mixed set solutions exhibit stability across both in-sample and out-of-sample scenarios.

Mixed set solutions do not always deliver the best solution compared to the parent sets for each s - t pair when we consider the best trade-offs among the performance measures; but when the mixed sets give better solutions than the parent sets, the margin by which they are better is much higher than the margin by which they are worse when they under-perform compared to the parent sets. This makes the average value of the solutions given by mixed sets over all the s - t pairs and scenarios better than the parent sets, and hence, make them a better option to find better trade-offs among the objective vector elements.

4. Conclusion

In this paper we have proposed a mixed uncertainty set approach to robust combinatorial optimization. Our results give a strong evidence in support of mixed sets giving superior solutions compared to individual approaches. This evidence paves way to further investigation into developing efficient algorithmic framework for building such mixed sets. For example, as an immediate future extension of this work, one option is to consider a Bayesian type approach, instead of a black-box optimizer like *irace*, which updates the weight on each parent set under consideration with every new bit of data collected. Furthermore, as the weights w_j assigned to each uncertainty set \mathcal{U}_j are not known precisely, an additional layer of uncertainty on \mathbf{w} may be added and handled using distributional robust optimization (see, e.g., [BTDHDW⁺13]).

Acknowledgements

We thank the anonymous referees, who provided valuable insight in the preparation of this paper.

References

- [ABV09] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European journal of operational research*, 197(2):427–438, 2009.
- [BGK18] Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.
- [BK18] Christoph Buchheim and Jannis Kurtz. Robust combinatorial optimization under convex and discrete cost uncertainty. *EURO Journal on Computational Optimization*, 6(3):211–238, 2018.
- [BL11] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.
- [BS03] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71, 2003.
- [BTDHDW⁺13] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenare, Bertrand Melenberg, and Gijis Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.
- [CDG19] André Chassein, Trivikram Dokka, and Marc Goerigk. Algorithms and uncertainty sets for data-driven robust shortest path problems. *European Journal of Operational Research*, 274(2):671–686, 2019.
- [CG18a] André Chassein and Marc Goerigk. Compromise solutions for robust combinatorial optimization with variable-sized uncertainty. *European Journal of Operational Research*, 269(2):544–555, 2018.

- [CG18b] André Chassein and Marc Goerigk. On scenario aggregation to approximate robust combinatorial optimization problems. *Optimization Letters*, 12(7):1523–1533, 2018.
- [CG18c] André Chassein and Marc Goerigk. Variable-sized uncertainty and inverse problems in robust optimization. *European Journal of Operational Research*, 264(1):17–28, 2018.
- [CH15] Trevor Campbell and Jonathan P How. Bayesian nonparametric set construction for robust optimization. In *American Control Conference (ACC), 2015*, pages 4216–4221. IEEE, 2015.
- [CKR18] Giovanni P Crespi, Daishi Kuroiwa, and Matteo Rocca. Robust optimization: Sensitivity to uncertainty in scalar and vector cases, with applications. *Operations Research Perspectives*, 5:113–119, 2018.
- [GMT14] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483, 2014.
- [GS16] Marc Goerigk and Anita Schöbel. Algorithm engineering in robust optimization. In *Algorithm engineering*, pages 245–279. Springer, 2016.
- [KZ10] Adam Kasperski and Paweł Zieliński. Minmax regret approach and optimality evaluation in combinatorial optimization problems with interval and fuzzy weights. *European Journal of Operational Research*, 200(3):680–687, 2010.
- [KZ16] Adam Kasperski and Paweł Zieliński. Robust discrete optimization under discrete and interval uncertainty: A survey. In *Robustness analysis in decision aiding, optimization, and analytics*, pages 113–143. Springer, 2016.
- [LIDLSB11] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The irace package, iterated race for automatic algorithm configuration. Technical report, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
- [Pos18] Michael Poss. Robust combinatorial optimization with knapsack uncertainty. *Discrete Optimization*, 27:88–102, 2018.
- [Wel10] Dominic JA Welsh. *Matroid theory*. Courier Corporation, 2010.

A. Additional Experimental Results

To understand how each uncertainty set performs for each s - t pair, we choose a representative configuration for each uncertainty set to compare the performance. The description of the configurations which are compared can be found below.

- Mixed Uncertainty Set: We use a mix of convex hull and ellipsoidal uncertainty. The scaling parameter, λ , and the weight parameter, w_j , of convex hull are 0.2234 and 0.7502 respectively. For the ellipsoidal set, the scaling and weight parameters take the values 5.4609 and 0.9796 respectively.

- Convex Hull: The scaling parameter is 0.2
- Ellipsoidal: The scaling parameter is 6.0
- Interval: The scaling parameter is 0.075

Figure 4 shows the comparison in performance for each measure (Avg, Max and CVaR), both in-sample and out-of-sample, for every pair of mixed set and parent set, i.e., we compare performance for mixed and convex hull sets, for mixed and ellipsoidal sets and for mixed and interval sets. This is achieved by taking the relative difference in the solutions delivered by the mixed set and the comparing parent set for each $s-t$ pair, i.e., we calculate the difference between the value of parent and mixed solution, and divide by the value of the mixed solution. While a negative value indicates that the solution delivered by the mixed set is better than the parent set, a positive value indicates the opposite. For the sake of clarity in the plots, we filter out the sample containing only zero values as they indicate that both the sets deliver exactly the same solution. In our case, for each pair of mixed and parent set, the number of $s-t$ pairs for which the solutions differed are as follows: 131 (21.8%) for mixed and convex hull sets; 91 (15.2%) for mixed and ellipsoidal sets; and 129 (21.5%) for mixed and interval sets.

Figures 4(a) and 4(d), and Figures 4(b) and 4(e) respectively compare the Avg and Max performance measures (both in-sample and out-of-sample) for each pair of mixed and parent set. For the majority of the $s-t$ pairs, mixed set performs better than both ellipsoidal and interval sets for both in-sample and out-of-sample data. Moreover, while in-sample performance given by convex hull is better than that of mixed set, the out-of-sample performance of convex hull is similar to that of mixed set. Similarly, for the CVaR performance measure, Figures 4(c) and 4(f) lead us to conclude that for majority of the $s-t$ pairs, for both in-sample and out-of-sample performances, while mixed set performs worse than ellipsoidal set, it performs better than the other two parent sets.

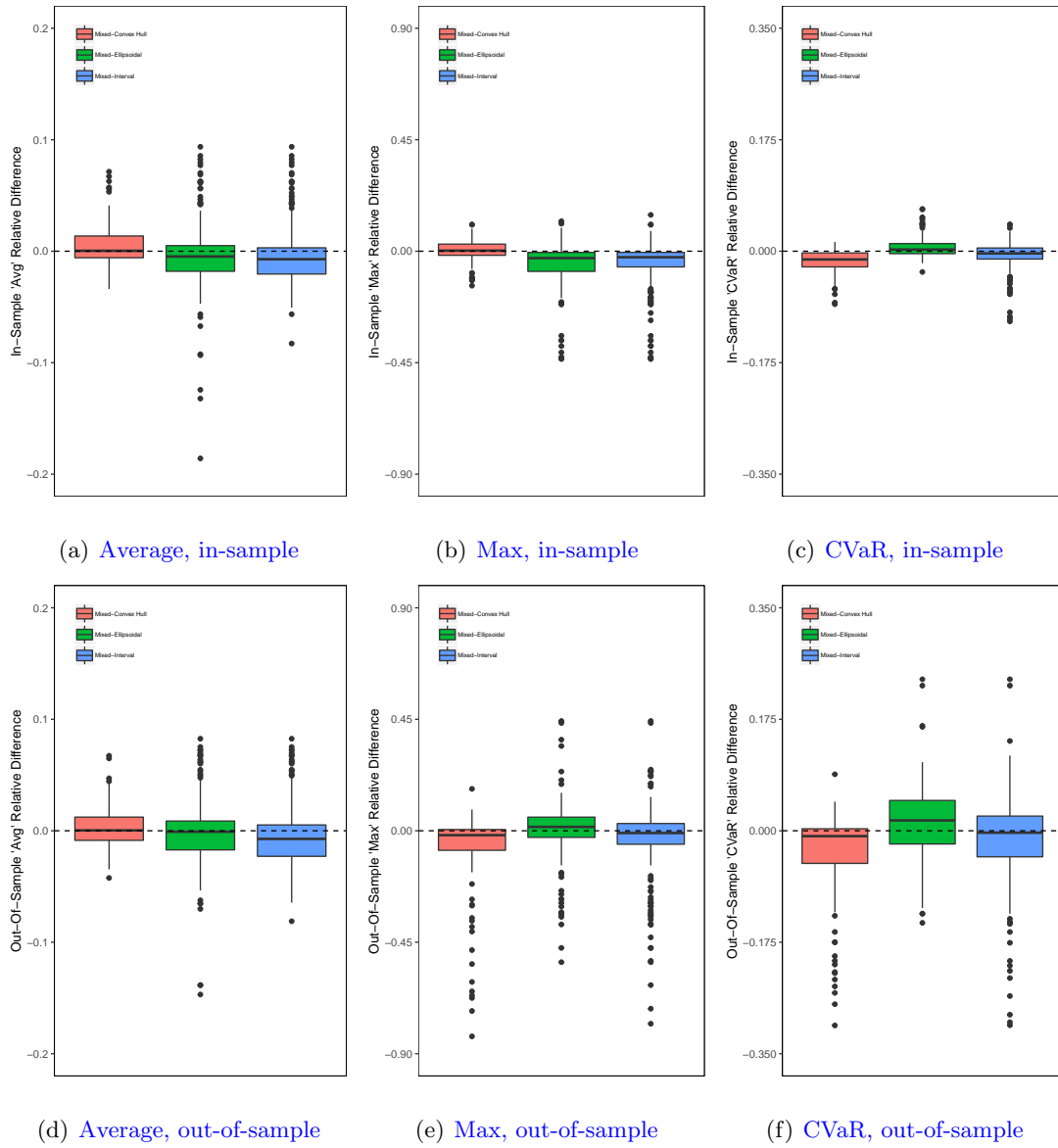


Figure 4: Relative differences with respect to average, worst-case (max), and average of worst 5% (CVaR) performance measures.