Solution Methods for Multi-Objective Robust Combinatorial Optimization

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

„Doctor rerum naturalium“

der Georg-August-Universität Göttingen

im Promotionsprogramm „PhD School of Mathematical Sciences“ (SMS)

der Georg-August University School of Science (GAUSS)

vorgelegt von

Lisa Thom

aus Kassel

Göttingen, 2018

Betreuungsausschuss

Prof. Dr. Anita Schöbel, Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen

Jun.-Prof. Dr. Anja Fischer, Juniorprofessur Management Science, Technische Universität Dortmund

Mitglieder der Prüfungskommission

Referentin: Prof. Dr. Anita Schöbel, Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen

Korreferentin: Dr. Marie Schmidt, Department of Technology and Operations Management, Erasmus University Rotterdam

Weitere Mitglieder der Prüfungskommission:

Jun.-Prof. Dr. Anja Fischer, Juniorprofessur Management Science, Technische Universität Dortmund

Prof. Dr. Gerlind Plonka-Hoch, Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen

Prof. Dr. Dominic Schuhmacher, Institut für Mathematische Stochastik, Georg-August-Universität Göttingen

Prof. Dr. Stephan Waack, Institut für Informatik, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung: 19. April 2018

Contents

1. Motivation

Applying classical optimization methods to real-world problems does not always yield the desired result. Two of the main difficulties are that often various (conflicting) objectives are relevant for the same problem and that not all parameters of a model can be predicted accurately in advance.

In many situations one does not pursue only one objective but has to balance several goals, which usually contradict each other: the best solution with respect to one criterion is rarely optimal considering all other criteria. For example, when driving on a road network and choosing between different routes, one might want to minimize travel time, fuel consumption and toll costs at the same time. However, the fastest route is rarely the most economical one regarding fuel consumption, and it is also more likely to contain toll roads.

This contradiction is sometimes resolved by assigning a weight to each criterion and optimizing the sum of the weighted objective functions. However, it is not always easy or even possible to find suitable weights in advance: to obtain an improvement in one objective, some impairment in another objective might or might not be tolerable, depending on the precise values. For example, to accept a 30 minutes delay in order to save some amount of toll cost might be a totally different consideration for an undelayed travel time of 15 minutes versus one of 15 hours.

On the other hand, given two routes with identical fuel consumption and toll cost, surely the faster one will be chosen, regardless of how the decision maker values the objectives. Therefore, in *multi-objective optimization*, one optimizes over a vector of objective functions instead of a single value. All solutions that cannot be improved in one objective without impairing another objective are of interest. They are called *(Pareto) efficient solutions*.

Furthermore, an obstacle often encountered when applying optimization methods in practice is missing information. Not all parameters of a model can be stated exactly in advance, in particular when predicting future developments. For example, when choosing a route in a road network, one cannot precisely predict the travel time and fuel consumption, because of potential traffic congestion, red traffic lights, weather conditions etc.

Uncertain problems can be tackled in several ways. To what extent perturbations in the parameters influence a given solution is analyzed by means of *sensitivity analysis*. In *stochastic optimization* the expected value, the variation or some other indicator based on the probability distribution is optimized, assuming that enough information on the probability of the various realizations of the data is given. *Robust optimiza-*

tion, on the other hand, hedges against (all) possible realizations of the uncertain data, called *scenarios*. For this purpose, information on possible scenarios but no probability information is assumed. For example, the range of the parameter values can be given as an interval: we might know that driving along a particular route takes between 15 and 30 minutes, but we don't know the expected travel time and variation. The information about the uncertain values can also be given in form of several distinct scenarios, e.g., weather scenarios or other events, which influence the traffic on some or all of the routes.

Intuitively, hedging against all scenarios means hedging against the worst case. Consequently, it is common to optimize the worst case objective value. For example, if an uncertain travel time is to be minimized, one chooses the tour whose worst possible duration is shortest. Nevertheless, there are also other interpretations of robustness, for example minimizing the worst case regret, where, given a specific scenario, the regret is the difference between the objective value of the chosen solution and the best possible objective value for this scenario.

Many real-world problems, as the problem of choosing a route, which we introduced above, do not yield only one but both of these obstacles. Imagine you want to choose a holiday destination and your objectives are the price, the time to get there and the activities you can take part in. The possible activities may depend on the weather, the travel time on traffic congestion or train delays and the price on foreign exchange rates or fuel costs. Another example occurs in the wood industry: Cutting a trunk into boards, one aims to maximize the revenue and minimize the waste. Both depend on the location of the core and damaged parts of the wood, which cannot be determined exactly from the outside, but only after the trunk has been cut.

The optimization problems considered in this thesis are combinatorial problems with multiple objectives and uncertain input parameters. We use concepts from the recently developed field of *multi-objective robust optimization*, which combines aspects of both multi-objective and robust optimization. Even though several concepts to define so-called *robust efficient solutions* have been developed during the last years, solution approaches are still rare. In this cumulative thesis, that is, in the underlying publications, we develop models and solution approaches for multi-objective robust combinatorial optimization problems based on techniques from both multi-objective and robust optimization.

In Chapter 2 we introduce concepts and methods of robust and multi-objective optimization as well as multi-objective robust optimization, including a brief literature review. The publications that constitute the cumulative part of this thesis are summarized in Chapter 3, followed by a discussion of the results in Chapter 4. The conclusion in Chapter 5 contains a summary of the results and potential aspects of future work.

2. Preliminaries and Related Literature

In this chapter, we introduce basic concepts and notations from multi-objective, robust and multi-objective robust optimization, and present related work.
In each of the sections we also devote one paragraph to combinatorial optimization within the scope of the respective field. In a combinatorial optimization problem, a set of elements E and a cost for each element is given, as well as a set of feasible subsets of E. Usually, the aim is to find a feasible subset, such that the sum of the contained elements' costs is minimal. An example is the shortest path problem, where E is the edge set in a graph and the feasible set consists of all simple paths between two given nodes.
Throughout the thesis we use the symbols $<$ (*smaller than*) and $\leqq$ (*smaller than or equal to*) to compare values in $\mathbb{R}$, in order to be consistent with the notation for comparing vectors, which we introduce in the next section (Definition 2.2). We write $A_{(i,\cdot)}$ for the i-th row of a matrix A and $A_{(\cdot,i)}$ for its i-th column. The transpose of a vector or matrix A is denoted by A^T.
Furthermore, we use a [. /.] notation to maintain a concise text: instead of writing "a feasible solution x is optimal if $z(x) \leqq z(y)$ for every feasible solution $y \neq x$ and uniquely optimal if $z(x) < z(y)$ for every feasible solution $y \neq x$" we write "a feasible solution x is [· / uniquely] optimal, if $z(x)[\leqq / <]z(y)$ for every feasible solution $y \neq x$".

2.1. Multi-Objective Optimization

The foundations of multi-objective optimization, also called multi-criteria optimization, were laid at the end of the 19th century by Edgeworth (1881) and Pareto (1896). For a recent textbook on the topic we refer to Ehrgott (2005). In order to optimize several (scalar-valued) objective functions simultaneously, each feasible solution is assigned an objective vector instead of a scalar objective value.

Definition 2.1. *Given a set $\mathcal{X}$ of feasible solutions and $k \in \mathbb{N}$ scalar-valued objective* *functions* $z_1, ..., z_k : \mathcal{X} \to \mathbb{R}$*, we call*

$$\min_{x \in \mathcal{X}} z(x) = \begin{pmatrix} z_1(x) \\ \vdots \\ z_k(x) \end{pmatrix}$$

a multi-objective optimization problem (MOP)*. For* $k = 1$ *we obtain a* single-objective *optimization problem.*

For $k \geqq 2$, a solution that minimizes all objectives at once does usually not exist. Therefore, we use the following relation to compare two vectors and to define *efficient* solutions, following the notation in Ehrgott (2005).

Definition 2.2. *Let* $k \in \mathbb{N}$*. For two vectors* $y^1, y^2 \in \mathbb{R}^k$ *we use the notation*

$$\begin{aligned} y^1 < y^2 &\Leftrightarrow y^1_i < y^2_i \text{ for all } i \in \{1, ..., k\}, \\ y^1 \leq y^2 &\Leftrightarrow y^1_i \leqq y^2_i \text{ for all } i \in \{1, ..., k\} \text{ and } y^1 \neq y^2, \\ y^1 \leqq y^2 &\Leftrightarrow y^1_i \leqq y^2_i \text{ for all } i \in \{1, ..., k\}. \end{aligned}$$

We also define the cones $\mathbb{R}^k_{[>/\geq/\geqq]} := \{y \in \mathbb{R}^k : 0[< / \leq / \leqq]y\}$*.*

By means of the relations in Definition 2.2 we define *(Pareto) efficient* solutions, which cannot be improved in one objective without worsening them in another objective, and the closely related concepts of *weakly* and *strictly* efficient solutions.

Definition 2.3. *A solution* $x \in \mathcal{X}$ *is a* [weakly/·/strictly] efficient solution *for MOP, if there does not exist any feasible solution* $x' \in \mathcal{X}, x' \neq x$ *with* $z(x')[< / \leq / \leqq]z(x)$*. Then* $z(x)$ *is called* [weakly/·/strictly] nondominated*. A* complete set *of efficient solutions is a set* $\mathcal{X}' \subseteq \mathcal{X}$ *such that for every efficient solution* x *there exists* $x' \in \mathcal{X}'$ *with* $z(x) = z(x')$*.*

Note that a solution $x \in \mathcal{X}$ is [weakly/·/strictly] efficient if and only if there is no $x' \in \mathcal{X}$ with $x' \neq x$ and

$$z(x') \in z(x) - \mathbb{R}^k_{[>/\geq/\geqq]}.$$

In contrast to single-objective optimization, where the optimal objective value is unique, there often exist many nondominated objective vectors if $k \geqq 2$. A common approach to find efficient solutions are *scalarization methods*: by solving a family of single-objective so-called *scalarized* problems, whose solutions are efficient for the multi-objective problem, one finds a set of solutions with several different (and possibly all) nondominated objective vectors. Ehrgott (2006) gives an overview on popular scalarization methods, among them the *weighted sum method* (e.g., Gass and Saaty, 1955), the *ϵ-constraint method* (Haimes et al., 1971; Chankong and Haimes, 1983) and the *weighted Chebychev method* (Bowman, 1976; Steuer and Choo, 1983).

Multi-Objective Combinatorial Optimization

Many combinatorial optimization problems have been extended to multi-objective combinatorial problems. An overview on multi-objective combinatorial optimization is given by Ehrgott and Gandibleux (2000) and Ehrgott (2005) among others. Often, there exist instances with exponentially many nondominated objective vectors, see, e.g., Hansen (1980) for the shortest path problem and Hamacher and Ruhe (1994) for the minimum spanning tree problem. Nevertheless, algorithms for solving particular single-objective combinatorial optimization problems can sometimes be extended to find all nondominated objective vectors of the multi-objective problem. For example, extensions of the famous labeling algorithms by Dijkstra (1959) and Bellman, Ford and Moore (e.g., Bellman, 1958) have been developed to solve the multi-objective shortest path problem (see, e.g., Martins, 1984; Corley and Moon, 1985; Paixão and Santos, 2013).

2.2. Robust Optimization

Robust optimization is one way to handle uncertain parameters in an optimization problem. No probability data is needed, but the potential realizations of the uncertain data are assumed to be given via an *uncertainty set* $\mathcal{U}$, which contains all possible *scenarios*.
In this thesis, if the feasible set of the optimization problem is subject to uncertainty, we aim to find solutions which are feasible for all scenarios, following seminal works on robustness, e.g., Soyster (1973) and Ben-Tal and Nemirovski (1998). For this purpose, the sets of feasible solutions under all scenarios can be intersected in advance to obtain a set of *robust feasible solutions*. Hence, in the following, we assume the feasible set $\mathcal{X}$ to be *deterministic*, which means that it is not subject to uncertainty, and define an uncertain optimization problem with uncertainty in the objective function only. Nevertheless, we also mention robustness concepts that do not inherently make this assumption.

Definition 2.4. *Given a feasible set of solutions $\mathcal{X}$, an uncertainty set $\mathcal{U}$, and an objective function $z : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, the family $(\mathcal{P}(\xi), \xi \in \mathcal{U})$ of optimization problems*

$$\mathcal{P}(\xi) \qquad \min_{x \in \mathcal{X}} z(x, \xi)$$

is called an uncertain optimization problem (UP)*. A problem that is not subject to uncertainty, e.g. UP with $|\mathcal{U}| = 1$, is called* deterministic*.*

Several *robustness concepts* have been developed to define *robust solutions* for UP. One of the most popular is *minmax robustness*, first introduced by Soyster (1973) and extensively studied, e.g., by Ben-Tal et al. (2009). A minmax robust optimal solution is a solution with minimal objective value in the *worst case*, i.e., it solves the following deterministic problem, called the *minmax robust counterpart* of the uncertain problem.

Definition 2.5. *Let an uncertain optimization problem UP be given. A solution $x \in \mathcal{X}$ is* minmax robust optimal *for UP, if it is optimal for the deterministic problem*

$$\min_{x\in\mathcal{X}} \sup_{\xi\in\mathcal{U}} z(x,\xi),$$

which is called minmax robust counterpart*.*

Other robustness concepts include *deviation robustness* (see Kouvelis and Yu, 1997), also called *minmax regret robustness*. Here, the maximal regret over all scenarios is minimized, which is the difference between the objective value of the respective solution and the optimal objective value for this scenario. If the optimal value is additionally used as a scaling factor for the regret, one obtains *relative robustness* (see Kouvelis and Yu, 1997). *Lightly robust solutions* (Fischetti and Monaci, 2009; Schöbel, 2014) are required to be not too bad in the most likely case, called *nominal scenario*. *Adjustable robustness* (Ben-Tal et al., 2004) or *recoverable robustness* (Cicerone et al., 2007; Liebchen et al., 2009; Erera et al., 2009) is used if part of the chosen solution can be determined or changed after the realization of the uncertain data. For an overview on robustness concepts see, e.g., Goerigk and Schöbel (2016).
Another approach to consider all scenarios at once, which we refer to as *multi-scenario optimality*, is inspired by (Pareto) efficiency in multi-objective optimization: one aims to find solutions which cannot be improved for one scenario without worsening them for another scenario. For the relationship between multi-scenario optimality and several robustness concepts see, e.g., Klamroth et al. (2017). Iancu and Trichakis (2014) combine multi-scenario efficiency and minmax robustness to define *Pareto robust optimal solutions*, which are both minmax robust optimal and multi-scenario optimal.
Apart from the robustness concept, the uncertainty set, too, plays an important role regarding the obtained solutions and the complexity of the robust problem. A *finite uncertainty set* contains a finite number of scenarios. In case of *interval uncertainty* the uncertain parameters vary independently of each other between given lower and upper bounds. Further common uncertainty sets include *ellipsoidal* and *polyhedral uncertainty sets*. Bertsimas and Sim (2003) introduced *bounded uncertainty*, also called *cardinality-constrained*, *budgeted*, *banded* or Γ*-uncertainty* (see also Bertsimas and Sim, 2004). They assume that the uncertain parameters vary independently of each other in given intervals, but not all of them deviate from their *nominal value*, which we assume here to be their minimal value.

Definition 2.6. *Let an uncertain optimization problem with $n \in \mathbb{N}$ uncertain parameters be given, with a nominal value $\hat{c}_j \in \mathbb{R}$ and an interval length $\delta_j \in \mathbb{R}_{\geqq}$ for each* *uncertain parameter c_j, where $j \in \{1, ..., n\}$. Further, let $\Gamma \in \mathbb{Z}$ with $0 \leqq \Gamma \leqq n$ be given. We define the* bounded uncertainty set *as*

$$\mathcal{U}^b := \left\{ c \in \mathbb{R}^n : c_j = \hat{c}_j + \beta_j \delta_j, \beta_j \in [0,1] \ \forall j \in \{1, ..., n\}, \sum_{j=1}^{n} \beta_j \leqq \Gamma \right\}.$$

Variations and extensions of bounded uncertainty have been developed, e.g., by Poss (2014) and Büsing and D'Andreagiovanni (2014).
Chassein et al. (2018) assume that the uncertainty set is determined based on a discrete sample of scenarios and experimentally investigate how different kinds of uncertainty sets influence the obtained minmax robust optimal solutions.

Robust Combinatorial Optimization

Robust combinatorial optimization problems have been investigated extensively, in particular with discrete and interval uncertainty, see, for example, Kouvelis and Yu (1997) and the recent survey by Kasperski and Zieliński (2016). When considering uncertainty in the objective function, the uncertain parameters are the costs of the elements. With discrete uncertainty, minmax robust counterparts of several polynomially solvable problems have been proven to be NP-hard, including the shortest path problem, the minimum spanning tree problem and the assignment problem (Murthy and Her, 1992; Kouvelis and Yu, 1997). If the costs of the elements vary independently of each other, e.g., in intervals, the minmax robust counterpart can be reduced to a deterministic problem by only considering the maximal cost of each element. For bounded uncertainty, Bertsimas and Sim (2003) have developed an algorithm to solve the minmax robust counterpart in polynomial time, provided that the underlying deterministic problem is polynomially solvable.

2.3. Multi-Objective Robust Optimization

The examples in Chapter 1 show that it is not uncommon for a real-world problem to be of multi-objective nature and to contain uncertain parameters, resulting in a multi-objective uncertain optimization problem.

Definition 2.7. *Given a feasible set of solutions $\mathcal{X}$, an uncertainty set $\mathcal{U}$, and a multi-objective function $z : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^k$, the family $(MOP(\xi), \xi \in \mathcal{U})$ of deterministic multi-objective optimization problems*

$$MOP(\xi) \qquad \min_{x \in \mathcal{X}} z(x, \xi)$$

is called a multi-objective uncertain optimization problem (MOUP)*.*

Remark 2.8. *Throughout this thesis we assume $\mathcal{X}$ and $\mathcal{U}$ to be compact and non-empty and the z_i to be continuous in x and ξ. In this case* $\max_{\xi \in \mathcal{U}} z_i(x, \xi)$ *exists for all $i \in \{1, ..., k\}$ and $x \in \mathcal{X}$.*

The field of multi-objective robust optimization, combining concepts and methods from robust and multi-objective optimization, has for the most part been developed during the last years and is currently gaining more and more interest. For a recent survey on multi-objective robust optimization see Wiecek and Dranichak (2016).

Robustness Concepts for Multi-Objective Optimization

Similar to single-objective robust optimization, several robustness concepts for multi-objective optimization have been introduced, which define *robust efficient solutions* for multi-objective uncertain optimization problems.
An intuitive approach to define robust efficient solutions for a multi-objective uncertain optimization problem is to choose solutions that are efficient for each scenario. It was first proposed by Bitran (1980) for linear problems with interval uncertainty and is often referred to as *necessary efficiency*. In terms of multi-objective robust optimization, it was established as *highly robust efficiency* by Kuhn et al. (2016) and Ide and Schöbel (2016).

Definition 2.9. *A solution $x \in \mathcal{X}$ is* highly robust efficient *for MOUP if*

$$\forall \xi \in \mathcal{U} \nexists x' \in \mathcal{X} : z(x', \xi) \leq z(x, \xi).$$

However, there is no guarantee that a highly robust efficient solution exists. Bitran (1980) propose a second reasonable criterion, often referred to as *possible efficiency*: the chosen solutions should be efficient for at least one of the scenarios. This concept is identical to *flimsily robust efficiency* by Kuhn et al. (2016) and Ide and Schöbel (2016).

Definition 2.10. *A solution $x \in \mathcal{X}$ is* flimsily robust efficient *for MOUP if*

$$\exists \xi \in \mathcal{U} \nexists x' \in \mathcal{X} : z(x', \xi) \leq z(x, \xi).$$

An extension of the single-objective concept of minmax robustness to multi-objective optimization was introduced by Kuroiwa and Lee (2012) (see also Fliege and Werner, 2014). They consider the worst case in each objective independently and search efficient solutions for the resulting deterministic multi-objective problem.

Definition 2.11. *Given a multi-objective uncertain optimization problem, we define*

$$\bar{z}(x) := \begin{pmatrix} \max_{\xi \in \mathcal{U}} z_1(x, \xi) \\ \vdots \\ \max_{\xi \in \mathcal{U}} z_k(x, \xi) \end{pmatrix}.$$

A solution $x \in \mathcal{X}$ *is* point-based minmax robust [weakly/$\cdot$/strictly] efficient *for MOUP, if it is a [weakly/$\cdot$/strictly] efficient solution for the multi-objective deterministic* robust counterpart $\min_{x\in\mathcal{X}} \bar{z}(x)$*, i.e., if there is no* $x' \in \mathcal{X}$ *with* $x' \neq x$ *and*

$$\bar{z}(x') \in \bar{z}(x) - \mathbb{R}^k_{[>/\geq/\geqq]}.$$

In the following, we abbreviate point-based minmax robust *to* pointMR.

This concept has been extensively applied, e.g., to portfolio optimization (Fliege and Werner, 2014), game theory (Yu and Liu, 2013) and the planning of sustainable supply chains (Hombach et al., 2017). Krüger et al. (2017) introduce the notion of a *robustness gap* for this concept, which measures what is lost by implementing a robust efficient solution instead of an efficient solution for a single scenario (see also Krüger, 2018a).
Since, in the concept of point-based minmax robust efficiency, the worst case is considered in each objective independently, the resulting worst case point can be arbitrarily far from the objective vectors obtained by evaluating each scenario. In contrast, the concept of *set-based minmax robust efficiency* (Ehrgott et al., 2014), takes the dependencies between the objectives into account by comparing the sets of objective vectors obtained for all scenarios (see also Avigad and Branke, 2008).

Definition 2.12. *Given a multi-objective uncertain optimization problem, we define the* outcome set *of a solution* $x \in \mathcal{X}$ *as*

$$z_{\mathcal{U}}(x) := \{z(x,\xi) : \xi \in \mathcal{U}\}.$$

A solution $x \in \mathcal{X}$ *is* set-based minmax robust [weakly/$\cdot$/strictly] efficient *for MOUP, if there exists no* $x' \in \mathcal{X}$ *with* $x' \neq x$ *and*

$$z_{\mathcal{U}}(x') \subseteq z_{\mathcal{U}}(x) - \mathbb{R}^k_{[>/\geq/\geqq]}.$$

In the following, we abbreviate set-based minmax robust *to* setMR.

This concept has been applied, e.g., to a veneer cutting problem (Ide et al., 2015) and the design of distributed energy supply systems (Majewski et al., 2017). Ide et al. (2014) generalize it to other cones than $\mathbb{R}^k_{[>/\geq/\geqq]}$.
Note that for $k = 1$ setMR efficiency and pointMR efficiency reduce to the single-objective concept of minmax robustness. Ehrgott et al. (2014) show the following connections between the two multi-objective concepts.

Lemma 2.13 (Ehrgott et al. (2014))**.** *Every pointMR [strictly/weakly] efficient solution is also setMR [strictly/weakly] efficient. In case of* objective-wise uncertainty, *i.e., if the uncertainty set can be written as* $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_k$ *and the uncertain problem as*

$$\left(\min_{x \in \mathcal{X}} \begin{pmatrix} z_1(x, \xi_1) \\ \vdots \\ z_k(x, \xi_k) \end{pmatrix}, \xi_i \in \mathcal{U}_i \ \forall \ i \in \{1, ..., k\} \right),$$

the sets of pointMR [weakly/·/strictly] efficient solutions and setMR [weakly/·/strictly] efficient solutions are identical.

The concepts of *convex hull efficiency* by Bokrantz and Fredriksson (2017) and *properly robust efficiency* by Kuroiwa and Lee (2012) are also based on the idea of minmax robustness. Other single-objective robustness concepts have also been transferred to multi-objective optimization, see Kuhn et al. (2016) and Ide and Schöbel (2016) for an extension of light robustness and Nikulin et al. (2013) for an extension of relative robustness.

Further concepts, including those by Gunawan and Azarm (2005); Deb and Gupta (2006); Witting et al. (2013), are also often called robustness concepts for multi-objective optimization, even though they do not follow the classical concepts of single-objective robust optimization and are sometimes more related to sensitivity analysis or stochastic optimization.

Botte and Schöbel (2016) consider a generalization of multi-scenario optimality and Pareto robust optimal solutions to the multi-objective case (see also Wiecek et al., 2009; Kuhn et al., 2016). In case of finitely many scenarios, they define *multi-scenario efficient solutions* as the efficient solutions to a deterministic multi-objective problem with one objective for each combination of a scenario and an original objective of the uncertain problem.

Definition 2.14. *Given a multi-objective uncertain optimization problem with finite uncertainty set* $\mathcal{U} = \{\xi_1, \ldots, \xi_m\}$*, a solution* $x \in \mathcal{X}$ *is* multi-scenario efficient *for MOUP if it is an efficient solution for*

$$\min_{x \in \mathcal{X}} \begin{pmatrix} z_1(x, \xi_1) \\ \vdots \\ z_1(x, \xi_m) \\ z_2(x, \xi_1) \\ \vdots \\ z_2(x, \xi_m) \\ z_3(x, \xi_1) \\ \vdots \\ z_k(x, \xi_m) \end{pmatrix}.$$

For an overview on different robustness concepts for multi-objective optimization we refer to Ide and Schöbel (2016) and Wiecek and Dranichak (2016).

Scalarization Methods for Multi-Objective Minmax Robust Optimization

To find pointMR efficient solutions, scalarization methods for multi-objective deterministic problems can be applied to the robust counterpart (see, e.g., Hassanzadeh et al., 2013; Kuroiwa and Lee, 2012; Fliege and Werner, 2014). In case of set-based minmax robust efficiency, the extension of scalarization methods is not as straightforward. Several methods to find setMR efficient solutions based on scalarizations have been developed. Ehrgott et al. (2014) introduce extensions of the weighted sum scalarization method and the ϵ-constraint method, which find setMR weakly efficient solutions. They show that the two methods do not always find the same solutions and that there can exist setMR efficient solutions, which cannot be found by either of these methods. A method based on the (augmented) weighted Chebyshev scalarization for finding setMR weakly efficient solutions has been introduced by Ide (2014). Bokrantz and Fredriksson (2017) consider order-preserving scalarizing functions $s : \mathbb{R}^k \to \mathbb{R}$ and the resulting scalarized problems $\min_{x \in \mathcal{X}} \max_{\xi \in \mathcal{U}} s(z(x, \xi))$. They show that for so-called *strongly increasing* scalarizing functions the solutions for the scalarized problem are setMR efficient. In an application they consider weighted p-norms as scalarizing functions, of which the weighted sum scalarization is a special case.

Schmidt et al. (2018) introduce the min-ordering and the max-ordering method, where a weighted minimum or maximum function is used as scalarizing function. That article is part of this thesis (see Addendum A.3) and is summarized in Section 3.3.

Uncertainty Sets

Finite and interval uncertainty sets have a straightforward equivalent in the multi-objective case. The idea of bounded uncertainty, however, can be extended to multiple objectives in different ways. It has first been extended to multi-objective problems with uncertainty only in the constraints (Doolittle et al., 2012). Hassanzadeh et al. (2013) consider an objective-wise uncertain linear problem with bounded uncertainty in each objective, i.e., with the following uncertainty set.

Definition 2.15. *Let a multi-objective uncertain optimization problem with $n \in \mathbb{N}$ uncertain parameters $\{c_{i,1}, \dots c_{i,n}\}$ in each objective function z_i be given. Further, let a nominal value $\hat{c}_{i,j} \in \mathbb{R}$ and an interval length $\delta_{i,j} \in \mathbb{R}_{\geqq}$ for each uncertain parameter $c_{i,j}$ be given as well as k numbers $\Gamma_1, ..., \Gamma_k \in \mathbb{Z}$ with $0 \leqq \Gamma_i \leqq n \ \forall i \in \{1, ..., k\}$. We define the* objective-wise bounded uncertainty set *as*

$$\mathcal{U}^{owb} := \Bigg\{ c \in \mathbb{R}^{k \times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0,1] \ \forall i \in \{1, ..., k\}, j \in \{1, ..., n\}, \\ \sum_{j \in \{1,...,n\}} \beta_{i,j} \leqq \Gamma_i \ \forall i \in \{1, ..., k\} \Bigg\}.$$

Schmidt et al. (2018) introduce another extension of bounded uncertainty, where they restrict the total number of uncertain parameters deviating from their nominal value, instead of regarding the objectives independently (see Addendum A.3 and its summary in Section 3.3).

Other Sources of Uncertainty

In this thesis, we only consider uncertainty arising from uncertain parameter values. However, there exist other possible reasons for uncertainty in optimization. Eichfelder et al. (2017) consider multi-objective optimization problems with *decision uncertainty* (see also Krüger, 2018a), which occurs when the decision variables cannot be implemented with accuracy. This concept is applied to a problem from agriculture in Krüger et al. (2018); Krüger (2018b). Doolittle et al. (2016) consider uncertainty arising when a scalarization method and scalarizing parameters are chosen in order to solve a deterministic multi-objective optimization problem. The survey by Wiecek and Dranichak (2016) contains an overview on sources of uncertainty in multi-objective optimization.

Multi-Objective Robust Combinatorial Optimization

An instance of a *multi-objective uncertain combinatorial optimization problem* (MOUCO) is given by a finite set $E = \{e_1, ..., e_n\}$, a feasible set $\mathcal{Q}$ containing subsets of E, and an uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{k \times n}$ containing all possible element costs: for every $c \in \mathcal{U}$, $c_{i,j}$ is the cost of element e_j w.r.t. the i-th objective.
One usually aims to minimize the sum of the contained elements' costs, i.e., MOUCO is the family $(\mathrm{MOCO}(c), c \in \mathcal{U})$ of multi-objective deterministic combinatorial problems

$$\mathrm{MOCO}(c) \qquad \min_{q \in \mathcal{Q}} z(q,c) \qquad \text{with} \quad z(q,c) := \begin{pmatrix} \sum\limits_{e_j \in q} c_{1,j} \\ \vdots \\ \sum\limits_{e_j \in q} c_{k,j} \end{pmatrix}$$

Alternatively, the set of feasible solutions can be written as a set of binary vectors $\mathcal{X} \subseteq \{0,1\}^n$, where each $x \in \mathcal{X}$ represents a feasible subset $q \in \mathcal{Q}$ with $x_j = 1 \Leftrightarrow e_j \in q$. Then, the objective function is defined by

$$z_i(x,c) := \sum_{j=1}^{n} c_{i,j} x_j \;\; \forall i \in \{1, ..., k\}.$$

Even though there exist several publications applying some robustness criterion to multi-objective uncertain combinatorial optimization problems (e.g., Mavrotas et al., 2015; Cintrano et al., 2017), their notions of robustness do not follow the definitions presented in this section, but are mostly based on concepts we rather associate with sensitivity analysis or stochastic optimization.

To the best of our knowledge, apart from the publications constituting the cumulative part of this thesis, only Kuhn et al. (2016) have developed solution approaches for multi-objective uncertain combinatorial problems applying some of the robustness concepts defined above. They confine their work to bi-objective problems with uncertainty in only one of the objective functions.

In the works summarized in Chapter 3 of this thesis (Raith et al., 2018b,a; Schmidt et al., 2018), multi-objective uncertain combinatorial optimization problems with any fixed number of uncertain objectives are considered, with a focus on shortest path problems. The authors develop approaches to find robust efficient solutions with respect to the concepts given in Definitions 2.9–2.12 and 2.14, considering finite, interval and bounded uncertainty sets.

3. Summary of the Publications

The cumulative part of this thesis consists of three research papers, which are summarized in this chapter. The author's own contribution to the respective manuscript is described at the end of each summary.
Section 3.1 summarizes the article Raith et al. (2018b), see Addendum A.1, which is published in the *European Journal of Operational Research.* The authors introduce two approaches to find pointMR efficient (or setMR efficient) solutions for multi-objective uncertain combinatorial optimization problems with objective-wise bounded uncertainty. From the general solution approaches they develop specific algorithms for the shortest path problem, which they compare experimentally.
The article Raith et al. (2018a), which is summarized in Section 3.2 and included in this thesis in Addendum A.2, is published in the journal *Networks.* So far, it has not been included in an issue, but the early view version is available online. In this paper, labeling algorithms for finding robust efficient solutions for the shortest path problem with a finite uncertainty set are developed, considering several different concepts of robust efficiency. Their performance is analyzed in an extensive numerical evaluation.
Section 3.3 contains a summary of the manuscript Schmidt et al. (2018), see Addendum A.3, which is available as preprint and has been submitted to the *European Journal of Operational Research* in January 2018. The authors introduce two scalarization methods for finding pointMR efficient or setMR efficient solutions for multi-objective uncertain optimization problems. They examine how the scalarized problems may be approached for combinatorial problems with particular uncertainty sets.

3.1. Multi-Objective Minmax Robust Combinatorial Optimization with Cardinality-Constrained Uncertainty

In Raith et al. (2018b), which we refer to as Publication 1, the authors consider multi-objective uncertain combinatorial problems with objective-wise bounded uncertainty, which they call cardinality-constrained uncertainty. They develop two approaches to find pointMR efficient (hence also setMR efficient) solutions: First they extend an algorithm for the single-objective minmax robust problem with bounded uncertainty to the multi-objective case with objective-wise bounded uncertainty. In addition, they provide an enhancement of the algorithm for one objective as well as a new proof of its validity, which they extend to prove the functionality of the multi-objective version. In the second approach, they transfer the multi-objective uncertain combinatorial optimization problem into a multi-objective deterministic optimization problem, whose efficient solutions form a superset of the robust efficient solutions for the original problem. They apply this approach to the shortest path problem by adjusting a labeling algorithm. Both algorithms are tested on a shortest path problem occurring in hazardous material transportation.

Note that the notation in Publication 1 differs slightly from the notation used in this thesis, e.g., the authors use $z^{\mathrm{R}}(q)$ instead of $\bar{z}(q)$ (see Definition 2.11) and $c_i(e_j)$ instead of $c_{i,j}$. In this summary, we use the notation introduced in Chapter 2.

Deterministic Subproblems Algorithm (DSA) for Single-Objective Problems

Bertsimas and Sim (2003) show that a minmax robust optimal solution for a single-objective uncertain combinatorial problem with bounded uncertainty can be found by solving $n+1$ deterministic problems, which we call *deterministic subproblems*. They assume that the elements in E and hence the indices of $\hat{c}, \delta$ are sorted with respect to the interval lengths, i.e., such that $\delta_1 \geqq \delta_2 \geqq \ldots \geqq \delta_n \geqq \delta_{n+1} := 0$. They define for every $l \in \{1, ..., n+1\}$ the problem

$$\mathcal{P}'(l) \quad \min_{q \in \mathcal{Q}} g^l(q) \qquad \text{with } g^l(q) := \sum_{e_j \in q} \hat{c}_j + \Gamma \cdot \delta_l + \sum_{\substack{e_j \in q, \\ j \leq l}} (\delta_j - \delta_l).$$

Since the summand $\Gamma \cdot \delta_l$ is solution-independent, every deterministic subproblem $\mathcal{P}'(l)$ can be interpreted as a combinatorial problem of the same type as the underlying problem with costs

$$c_j^l := \begin{cases} \hat{c}_j + (\delta_j - \delta_l) & \text{for } j < l \\ \hat{c}_j & \text{else.} \end{cases} \tag{3.1}$$

The *Deterministic Subproblems Algorithm (DSA)* solves $\mathcal{P}'(l)$ for $l = 1, ..., n+1$ and chooses among the obtained solutions the solution with minimal objective value. Hence, in case the underlying deterministic problem is polynomially solvable (e.g., the shortest path problem or the minimum spanning tree problem), the DSA has polynomial runtime.

Bertsimas and Sim (2003) prove with help of dualization that the DSA indeed finds a minmax robust optimal solution. In Publication 1, the authors introduce an alternative proof, which they later extend to prove their algorithm for the multi-objective case: They show that $g^l(q) \geqq \max_{c \in \mathcal{U}^b} z(q, c) \; \forall q \in \mathcal{Q}, l \in \{1, ..., n+1\}$ and that for each $q \in \mathcal{Q}$ there exists $\tilde{l} \in \{1, \ldots, n+1\}$ such that $g^{\tilde{l}}(q) = \max_{c \in \mathcal{U}^b} z(q, c)$. Therefore, every minmax robust optimal solution is optimal for at least one deterministic

subproblem $\mathcal{P}'(\tilde{l})$, whose optimal objective value is smaller than or equal to the optimal objective values of the other subproblems.
According to the results by Bertsimas and Sim (2003); Park and Lee (2007); Lee and Kwon (2014), the number of subproblems to be solved can be reduced to $\lceil \frac{n-\Gamma}{2} \rceil + 1$. The authors of Publication 1 show that, in addition, a subproblem needs not to be solved, if the solution of an already solved subproblem has the following property.

Lemma 3.1 (Publication 1, Lemma 9)**.** *Let $1 \leqq \tilde{l} < l \leqq |E| + 1$ and let $q^{\tilde{l}}$ be an optimal solution for $\mathcal{P}'(\tilde{l})$. If $q^{\tilde{l}}$ does not contain any of the elements $e_1, ..., e_{l-1}$, then it is optimal for $\mathcal{P}'(l)$.*

The authors point out that even though this result does not improve the theoretical worst case runtime, their experimental evaluation shows its use for practical applications.

DSA for Multi-Objective Problems

For the validity of the DSA for single-objective problems it is crucial that the elements in E, and hence the indices of $\hat{c}, \delta$, are sorted such that the entries of δ are decreasing. However, the authors of Publication 1 point out that in the multi-objective case a respective order of the elements does not necessarily exist: if they are sorted such that the interval lengths in the first objective are decreasing, i.e., $\delta_{1,1} \geqq \delta_{1,2} \geqq \ldots \geqq \delta_{1,n}$, the interval lengths in the other objectives are not necessarily decreasing as well. Therefore, given a multi-objective uncertain combinatorial optimization problem with objective-wise bounded uncertainty, deterministic subproblems cannot be defined analogous to the single-objective case. However, the authors define suitable multi-objective deterministic subproblems in a similar way: For each $l = (l_1, \ldots, l_k) \in L := \{1, ..., n+1\} \times ... \times \{1, ..., n+1\}$ they define

$$(\mathcal{MOP}'(l)) \quad \min_{q \in \mathcal{Q}} g^l(q) \qquad \text{with } g^l(q) := \begin{pmatrix} \sum\limits_{e_j \in q} \hat{c}_{1,j} + \Gamma_1 \cdot \bar{\delta}^1_{l_1} + \sum\limits_{e_j \in q \cap E^1_{l_1}} (\delta_{1,j} - \bar{\delta}^1_{l_1}) \\ \vdots \\ \sum\limits_{e_j \in q} \hat{c}_{k,j} + \Gamma_k \cdot \bar{\delta}^k_{l_k} + \sum\limits_{e_j \in q \cap E^k_{l_k}} (\delta_{k,j} - \bar{\delta}^k_{l_k}) \end{pmatrix},$$

where, for every $i \in \{1, ..., k\}$ and $l_i \in \{1, ..., n\}$, $E^i_{l_i} \subseteq E$ contains a set of l_i elements with largest interval lengths w.r.t. the i-th objective, i.e., $|E^i_{l_i}| = l_i$ and

$$\delta_{i,j} \geqq \delta_{i,j'} \ \forall e_j \in E^i_{l_i}, e_{j'} \notin E^i_{l_i}.$$

Further, for all $i \in \{1, ..., k\}$, they define $E^i_{n+1} := E, \bar{\delta}^i_{n+1} := 0$ and

$$\bar{\delta}^i_{l_i} := \min_{e_j \in E^i_{l_i}} \delta_{i,j} \ \forall l_i \in \{1, ..., n\},$$

hence, $\bar{\delta}^i_{l_i}$ equals the l_i-largest of the interval lengths w.r.t. the i-th objective. Note that $E^i_{l_i}$ and $\bar{\delta}^i_{l_i}$ are not variables, but can be precomputed.
Here, the efficient solution of the subproblems can be found by solving a multi-objective deterministic combinatorial problem of the same type as the underlying problem with costs

$$c^l_{i,j} := \begin{cases} \hat{c}_{i,j} + (\delta_{i,j} - \bar{\delta}^i_{l_i}) & \text{for } e_j \in E^i_{l_i} \\ \hat{c}_{i,j} & \text{else.} \end{cases}$$

The authors propose an algorithm (Algorithm 3 in Publication 1), referred to as *DSA*: First, it searches a complete set OPT^l of efficient solutions for $\mathcal{MOP}'(l)$ for every $l \in L$. It then returns all $q \in \bigcup_{l \in L} OPT^l$ for which there exists no $q' \in \bigcup_{l \in L} OPT^l$ with $\bar{z}(q') \leq \bar{z}(q)$. The authors prove that the DSA indeed finds a complete set of efficient solutions for $\min_{q \in \mathcal{Q}} \bar{z}(q)$ (Publication 1, Theorem 10), because $g^l(q) \geqq \bar{z}(q)$ for all $q \in \mathcal{Q}, l \in L$ and for each $q \in \mathcal{Q}$ there exists $\tilde{l} \in L$ such that $\bar{z}(q) = g^{\tilde{l}}(q)$. Note that the found solutions are both pointMR efficient and setMR efficient, because the problem is objective-wise uncertain (see Definition 2.11 and Lemma 2.13).
The authors show that the number of subproblems to be solved can be reduced to $\prod_{i=1}^k \left(\left\lceil \frac{|E| - \Gamma_i}{2} \right\rceil + 1 \right)$ (Publication 1, Lemma 12), using the results for the single-objective problem by Bertsimas and Sim (2003); Park and Lee (2007); Lee and Kwon (2014). Furthermore, a result similar to Lemma 3.1 can be used to skip some of these subproblems, if the solutions of a formerly solved subproblem fulfill a special condition (Publication 1, Lemma 13). We refer to this method as *solution checking.*
In addition, the authors show that the number of subproblems to be solved can further be reduced significantly, if the problem has *[partly/·] objective-independent element order*, i.e., if the elements can be ordered such that the interval lengths are in decreasing order for [several/all] objectives and the respective Γ_i are identical (Publication 1, Lemma 17). In case of objective-independent element order, $\left\lceil \frac{|E| - \Gamma_{i_1}}{2} \right\rceil + 1$ subproblems suffice.

Bottleneck Approach

The authors present a second solution approach, where the multi-objective uncertain problem is transformed to a multi-objective deterministic problem, whose set of efficient solutions contains a complete set of efficient solutions for $\min_{q \in \mathcal{Q}} \bar{z}(q)$.
For this, they use the following notation for the h-greatest interval length in a solution $q \in \mathcal{Q}$ w.r.t. a given $i \in \{1, ..., k\}$ (see Publication 1, Definition 18): for a subset $q \subseteq E$ and given interval lengths $\delta_{i,j}$ for all $e_j \in E$, they sort the elements in q by decreasing interval lengths and define $h\text{-}\max_{e_j \in q} \delta_{i,j}$ as the interval length of the h-th element according to this sorting.
They first explain their approach for single-objective problems and then extend it

to the multi-objective case. For a given MOUCO with k objectives they define a multi-objective deterministic problem with $\sum_{i=1}^{k}(\Gamma_i + 1)$ objectives:

$$\text{MODCO} \quad \min_{q \in \mathcal{Q}} z^{\mathbf{D}}(q) \qquad \text{with} \quad z^{\mathbf{D}}(q) := \begin{pmatrix} \sum_{e_j \in q} \hat{c}_{1,j} \\ \max_{e_j \in q} \delta_{1,j} \\ 2\text{- } \max_{e_j \in q} \delta_{1,j} \\ \vdots \\ \Gamma_1\text{- } \max_{e_j \in q} \delta_{1,j} \\ \sum_{e_j \in q} \hat{c}_{2,j} \\ \max_{e_j \in q} \delta_{2,j} \\ \vdots \\ \Gamma_k\text{- } \max_{e_j \in q} \delta_{k,j} \end{pmatrix}.$$

They show that every pointMR efficient solution for MOUCO is an efficient solution for MODCO and that a complete set of efficient solutions for MODCO contains a complete set of efficient solutions for $\min_{q \in \mathcal{Q}} \bar{z}(q)$ (Publication 1, Theorem 22).

Label Setting Algorithm (LSA) for the Multi-Objective Uncertain Shortest Path Problem

To use the bottleneck approach, one needs an algorithm to find a complete set of efficient solutions for MODCO. In Publication 1, the authors introduce such an algorithm for the multi-objective uncertain shortest path problem with non-negative edge lengths, where E is the edge set of a graph and $\mathcal{Q}$ is the set of all simple paths from a start node s to a termination node t. They adjust the label setting algorithm of Martins (1984) for the multi-objective deterministic shortest path problem. The structure of the algorithm is the same as that of the algorithm of Martins: A *label* at a node v represents a path q from s to v. It has a cost vector $y(l)$, equal to the cost of q, and a predecessor label l' at the predecessor node v' of v on q, representing the subpath of q from s to v'. Starting with a *temporary* label of cost 0 at s, as long as there exists at least one temporary label, the algorithm

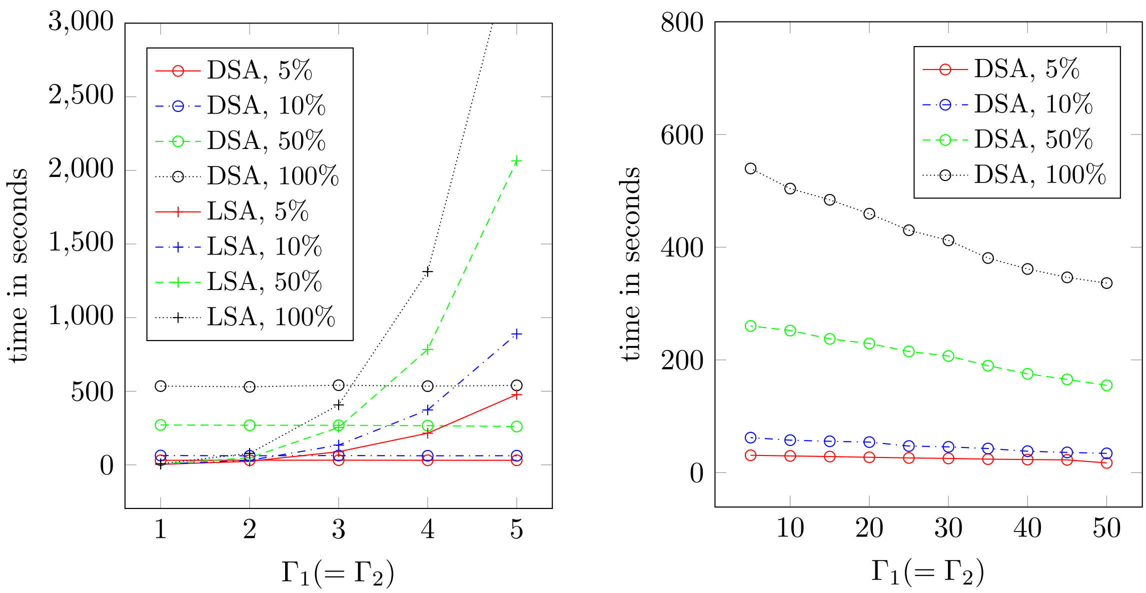
1. chooses a temporary label l' at a node v' to make it *permanent* instead of temporary,
2. produces new temporary labels at the end of the outgoing edges of v', whose predecessor label is l',
3. deletes every temporary label l for which a label $\tilde{l}$ at the same node with $y(\tilde{l}) \leq y(l)$ exists.

For the classical multi-objective shortest path problem, where each objective is the sum of the edge costs w.r.t. this objective, the cost $y(l)$ of a new label is obtained by

adding the cost of the predecessor label $y(l')$ to the cost of the last edge $e_j := (v', v)$. In Publication 1, the authors define a new procedure in order to obtain suitable label costs for MODCO: They add the nominal costs of e_j to the components of $y(l)$ corresponding to the sum objectives in MODCO. The interval lengths $\delta_{i,j}$ associated to e_j are compared to the other components of $y(l)$ and inserted at the right place (Publication 1, Algorithm 6).
In Step 1, the algorithm in Publication 1 chooses the label with the smallest aggregated costs, as proposed by Iori et al. (2010). It also differs from the Algorithm of Martins in Step 3: if several labels with the same costs at the same node exist, all but one of them are deleted, because the aim is to find a complete set of efficient solutions, not all efficient solutions.
The authors show that the adjusted labeling algorithm indeed finds a complete set of efficient solutions for MODCO (Publication 1, Theorem 27) and propose an additional filtering step to obtain a complete set of efficient solutions for $\min_{q \in \mathcal{Q}} \bar{z}(q)$ (see Publication 1, Algorithm 7 and Corollary 28). In the following the entire algorithm, including the filtering step, is called *LSA*.

Experimental Evaluation

The authors compare the performance of the two algorithms DSA and LSA for a multi-objective uncertain shortest path problem arising from hazardous material transportation. The aim is to find a path in a road network that minimizes travel time on the one hand and the population affected by the hazardous material in a potential accident on the other hand. Both objectives are uncertain, because the travel time depends, for example, on traffic congestion and the population in the area is influenced, e.g., by local events or regular shifts in population during the work day. The travel time intervals are obtained via an iterative algorithm to solve a traffic assignment problem. The population interval lengths are chosen randomly up to a given percentage of the assigned nominal values. Varying this percentage, referred to as *population uncertainty*, several different instances are constructed.
All methods to reduce the number of subproblems of the DSA, which are described above, are implemented. The subproblems of the DSA are solved with an implementation of the algorithm of Martins (1984) with the same adjustments in Steps 1 and 3 as in the LSA: in Step 1 the temporary label with the smallest aggregated cost is chosen and in Step 3, if there exist labels with the same cost at the same node, all but one are deleted.
The results show that the minimal number of robust efficient solutions in a complete set and the runtime of both algorithms generally increases with increasing population uncertainty. Comparing the performance of the two algorithms, the authors observe that with increasing values of Γ_i the runtime of the DSA decreases, whereas the runtime of the LSA increases (see Figure 3.1). This can be explained by the decreasing number of subproblems of the DSA and the increasing number of objec-



(a) $\Gamma_1 = \Gamma_2 \in \{1, ..., 5\}$ (b) $\Gamma_1 = \Gamma_2 \in \{5, 10, ..., 50\}$

Figure 3.1.: Running time of the DSA and the LSA for several values of Γ_i and population uncertainty on two different scales (Publication 1, Figure 5).

tives of MODCO, which is solved by the LSA. Indeed, for very small values of Γ_i the LSA solves the given instances faster than the DSA, whereas the DSA has a better performance for higher values of Γ_i. This is also true if the problem has three objectives instead of two objectives, which is tested on instances with a third (artificial) objective.

The authors further generate an instance with two strongly correlated objective functions, using the travel time as one objective and constructing a second objective by multiplying the nominal times and the interval lengths each by a random factor between 0.9 and 1.1. Both algorithms benefit from the correlation in terms of runtime, but the LSA benefits more: while for $\Gamma_1 = \Gamma_2 = 4$ the DSA already performed better for all tested instances with two uncorrelated objective functions, the LSA solved the correlated instances faster than the DSA up to $\Gamma_1 = \Gamma_2 = 26$.

In addition to the comparisons of the DSA and the LSA, the authors investigate the effect of the proposed enhancements: First, they compare the performance of the DSA with solution checking to a version without solution checking. The results show that the algorithm is accelerated substantially if subproblems can be skipped in this way, and it is not significantly slowed down by the procedure even if no subproblems can be skipped. Second, they test the DSA on an instance with objective-independent element order. They compare the performance of the DSA for general instances to a special version *DSA-oi*, which takes into account that the number of subproblems can be reduced further in case of objective-independent element order (Publication 1,

Lemma 17). As expected, the DSA-oi solves this instance much faster than the general version of the DSA. They also implement a procedure to check whether an instance has objective-independent element order, which does not take much time in comparison to the total running time of the DSA.
The authors conclude that the DSA solves most of the tested instances faster, but that the LSA performs better for small values of Γ_i, in particular if the objectives are strongly correlated. When implementing the DSA, they recommend to use the proposed enhancements and to check whether the special version for instances with (partial) objective-independent element order can be used, because the additional procedures do not take much time in comparison to the total running time and, if subproblems can be skipped, the algorithm is accelerated significantly.

Own Contribution

This article is joint work with Andrea Raith, Marie Schmidt and Anita Schöbel. The ideas leading to this publication were developed cooperatively by all four authors. Most of the details, including the algorithms, the technicalities in the proofs and the examples, were contributed by myself, of course with consultation of the other authors. I have done approximately half of the implementations and the main part of the experiments. Most of the text and figures, both in the theoretical and the experimental part, were produced by myself.

3.2. Extensions of Labeling Algorithms for Multi-Objective Uncertain Shortest Path Problems

This section summarizes the article Raith et al. (2018a), which we refer to as Publication 2. In this paper, the authors consider the multi-objective uncertain shortest path problem with finite uncertainty. They aim to find multi-scenario efficient, flimsily, highly, point-based minmax and set-based minmax robust efficient solutions. First, they analyze why it is, for most of the considered concepts, not straightforward to use labeling algorithms for the multi-objective uncertain problem. They then develop algorithms to find robust efficient solutions, by either extending a generic multi-objective label correcting algorithm or using it repeatedly. In a numerical study, the authors analyze and compare the performance of the developed algorithms on two different types of networks.

An instance of the *multi-objective uncertain shortest path problem* (MOUSP) is given by a graph $G = (V, E)$ with node set V and edge set E, a start node $s \in V$, an end node $t \in V$ and an uncertainty set $\mathcal{U} \subseteq \mathbb{R}^{k \times n}$, containing all possible edge costs.

For every $v \in V$, let $\mathcal{Q}^v$ denote the set of all simple paths from s to v. MOUSP is then a special case of MOUCO with element set E, feasible set $\mathcal{Q}^t$ and uncertainty set $\mathcal{U}$ (see page 12). In this publication, finite uncertainty sets are considered, i.e., $\mathcal{U} = \{c^1, c^2, \dots, c^r\}$ for some $r \in \mathbb{N}$.

To keep a consistent notation throughout the thesis, the notation in this summary differs from the notation in the article itself. At some places we point out the original notation, to allow an easier understanding when looking something up in the article. For example, in the notation of Publication 2, the uncertainty set $\mathcal{U}$ is given as a set of scenarios $\{\xi_1, \dots, \xi_r\}$ and the costs as a function $c : \mathcal{Q} \times \mathcal{U} \to \mathbb{R}^k$, where $c_i(e_j, \xi_d)$ is identical to $c^d_{i,j}$ in our notation. Hence, an instance of MOUSP is given as $(G, \mathcal{U}, c, s, t)$ in Publication 2 instead of $(G, \mathcal{U}, s, t)$ in our notation.

General Label Correcting Algorithm

The authors consider a generic multi-objective label correcting algorithm with label selection method (see, e.g., Guerriero and Musmanno, 2001), called *Algorithm 1.* A *label* at a node v represents a path q from s to v. It has a cost $z(l)$, which equals the cost of q, and a predecessor label l' at the predecessor node v' of v on q, representing the subpath of q from s to v'.
The label correcting algorithm starts with an empty label set $\mathcal{L}$ and a second label set $\mathcal{T}$ containing a label of cost 0 at node s. As long as $\mathcal{T}$ is not empty, the algorithm

1. chooses a label l' in $\mathcal{T}$ at a node v' and moves it to the label set $\mathcal{L}$ instead,

2. produces new labels at the end of the outgoing edges of v', whose predecessor label is l',

3. adds every new label l to $\mathcal{T}$, if there exists no label $\tilde{l} \in \mathcal{T} \cup \mathcal{L}$ at the same node that has identical cost or dominates l,

4. deletes every label $\tilde{l} \in \mathcal{T} \cup \mathcal{L}$ that is dominated by a new label $l \in \mathcal{T}$ at the same node.

Afterwards, it returns all labels in $\mathcal{L}$ at t.

In the multi-objective deterministic case, the cost of a path q, i.e., the cost of the label l representing q, is the sum of the cost vectors of the edges in q. To compute $z(l)$, one adds the cost of its predecessor label to the cost of the last edge in q. A label l dominates another label $\tilde{l}$, if $z(l) \leq z(\tilde{l})$. When the algorithm stops, the labels at t represent a complete set of efficient paths from s to t.
The authors point out that in the uncertain case with finite uncertainty set, the cost

of an edge e_j can be written as a matrix

$$Z(e_j) := \begin{pmatrix} c^1_{1,j} & c^2_{1,j} & \dots & c^r_{1,j} \\ c^1_{2,j} & c^2_{2,j} & \dots & c^r_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ c^1_{k,j} & c^2_{k,j} & \dots & c^r_{k,j} \end{pmatrix} \in \mathbb{R}^{k \times r},$$

the cost of a path as $Z(q) := \sum_{e \in q} Z(e)$ and the cost of a label l representing q as $Z(l) := Z(q)$. (In the original paper, $[Z(e)/Z(q)/Z(l)]$ is denoted as $[c(e)/z(q)/z(l)]$.) For a given concept of robust efficiency, they define a complete set of robust efficient paths as a set of robust efficient paths $\mathcal{Q}' \subseteq \mathcal{Q}^t$, such that for each robust efficient path q there exists $q' \in \mathcal{Q}'$ with $Z(q) = Z(q')$.
Using cost matrices instead of cost vectors, the label setting algorithm can easily be transferred to the uncertain case, if a suitable definition of dominance is given. The authors show that a complete set of robust efficient solutions can be found with a straightforward transfer of Algorithm 1, referred to as *Algorithm 1'*, if the concept of robust efficiency fulfills the following two conditions (see Publication 2, Theorem 8).

1. Principle of optimality: For every instance $(G, \mathcal{U}, s, t)$ of MOUSP we require: if $q \in \mathcal{Q}^t$ is a robust efficient path for $(G, \mathcal{U}, s, t)$, then for every node v in q its subpath $q_{s,v}$ from s to v is robust efficient for the instance $(G, \mathcal{U}, s, v)$.

2. For every $k, r \in \mathbb{N}$ there exists a binary (dominance) relation $R \subseteq \mathbb{R}^{k \times r} \times \mathbb{R}^{k \times r}$ with the following properties:

   a) The relation is consistent with the concept of robust efficiency: for all instances with k objectives and $|\mathcal{U}| = r$:

   $$q \in Q^t \text{ is robust efficient } \Leftrightarrow \nexists\, q' \in Q^t : (Z(q'), Z(q)) \in R$$

   b) Domination property: For all instances with k objectives and $|\mathcal{U}| = r$:

   $$\begin{aligned} &q \in Q^t \text{ is not robust efficient} \\ &\Rightarrow \exists \text{ robust efficient } q' \in Q^t : (Z(q'), Z(q)) \in R \end{aligned}$$

   c) R is transitive, i.e., $(Y^1, Y^2) \in R, (Y^2, Y^3) \in R \Rightarrow (Y^1, Y^3) \in R$.

   We say that q' *dominates* q if $(Z(q'), Z(q)) \in R$.

Condition 2 defines the notion of dominance used in Steps 3 and 4. Further, the instance needs to be *conservative*, i.e., the cost of every circle C in G is either 0 or we have $(Y, Y + Z(C)) \in R$ for all $Y \in \mathbb{R}^{k \times r}$.

Labeling for the Multi-Objective Robust Shortest Path Problem

The authors investigate whether the considered concepts (multi-scenario efficiency, flimsily and highly robust efficiency, pointMR and setMR efficiency) fulfill the two conditions given above. In case any of the conditions is not fulfilled, they propose algorithms to nevertheless find a complete set of robust efficient solutions. These algorithms are either an extension of Algorithm 1' (*extended labeling algorithms*) or solve Algorithm 1' several times for auxiliary problems with different definitions of dominance and compute the solution set from the obtained solutions for the auxiliary problems (*repeated labeling algorithms*).

For multi-scenario efficiency with finite uncertainty set, MOUSP reduces to a multi-objective deterministic problem (see Definition 2.14). Hence, one can directly use the label correcting algorithm for multi-objective deterministic problems, Algorithm 1.

For flimsily robust efficiency (see Definition 2.10), the authors show that the principle of optimality (Condition 1) is fulfilled, but that no dominance relation exists with Property 2a as required in Condition 2.
They introduce an extension of Algorithm 1' to nevertheless find a complete set of flimsily robust efficient solutions (EL-Flimsily), see Algorithm 2 in Publication 2. For each label l, an additional vector $x(l) \in \{0,1\}^r$ is stored, which is set to 0 in the beginning of the algorithm. When comparing two labels $l, \tilde{l}$ in Step 3 or Step 4, one compares them for each scenario independently, i.e., the cost matrices are compared column-wise. If $Z(l)_{(\cdot,d)} \leq Z(\tilde{l})_{(\cdot,d)}$, then the entry $x_d(\tilde{l})$ is set to 1, indicating that the label $\tilde{l}$ is *dominated in scenario* c^d. A label is deleted if it is dominated in every scenario. The authors show that EL-Flimsily indeed finds a complete set of flimsily robust efficient solutions, if the instance fulfills a condition similar to conservativeness (Publication 2, Theorem 12).
The authors also propose a repeated labeling algorithm (RL-Flimsily), see Algorithm 3 in Publication 2. For each $c^d \in \mathcal{U}$ it executes Algorithm 1' with the following definition of dominance: a label l dominates another label $\tilde{l}$ at the same node, if $Z(l)_{(\cdot,d)} \leq Z(\tilde{l})_{(\cdot,d)}$. Afterwards, the union of the obtained solution sets is returned.

The authors show further that for highly robust efficiency (see Definition 2.9), Condition 1 is fulfilled, too, but no dominance relation exists with Property 2b as required in Condition 2. Since the highly robust efficient solutions are exactly the flimsily robust efficient solutions that are not dominated in any scenario, the authors propose an algorithm (EL-Highly) that executes EL-Flimsily and returns only those labels l with $x(l) = 0$ (Publication 2, Algorithm 4). As an alternative they propose a repeated labeling algorithm (RL-Highly), which works similar to RL-Flimsily, but returns the intersection of the obtained solution sets instead of their union (Publication 2, Algorithm 5).

Finally, the authors show that suitable dominance relations exist for pointMR and setMR efficiency. For pointMR efficiency that is R^{point} defined as

$$(Y, Y') \in R^{\text{point}} \Leftrightarrow \begin{pmatrix} \max_{d=1,\ldots,r} Y_{1,d} \\ \vdots \\ \max_{d=1,\ldots,r} Y_{k,d} \end{pmatrix} \leq \begin{pmatrix} \max_{d=1,\ldots,r} Y'_{1,d} \\ \vdots \\ \max_{d=1,\ldots,r} Y'_{k,d} \end{pmatrix}$$

and for setMR efficiency that is R^{set} given by

$$(Y, Y') \in R^{\text{set}} \Leftrightarrow \bigcup_{d=1,\ldots,r} \{Y_{(\cdot,d)}\} \subseteq \bigcup_{d=1,\ldots,r} \{Y'_{(\cdot,d)}\} - \mathbb{R}^k_{\geq}.$$

On the other hand, both concepts do not fulfill Condition 1, i.e., subpaths of robust efficient paths are not necessarily robust efficient themselves.
To overcome this obstacle, the authors adopt an idea introduced by Yu and Yang (1998); Kouvelis and Yu (1997) for the single-objective case with integer edge costs, and introduce an extended labeling algorithm to find pointMR and setMR efficient solutions (Publication 2, Algorithm 6).
Assuming integer edge costs, each label l at a node v is assigned a *prediction matrix* $A(l) \in \mathbb{Z}^{k\times r}$ (denoted $a(l)$ in Publication 2), which contains assumed costs for continuing the represented path from v to t. Hence, every path from s to v can be represented by several labels with different prediction matrices.
The algorithm starts with several labels of cost 0 at s, one for every prediction matrix with integer entries that lie between precomputed lower and upper bounds. When generating a new label, its prediction matrix is obtained by subtracting the last edge's cost from the prediction matrix of its predecessor label. In steps 3 and 4 only those labels are compared, whose prediction matrices are identical. A label l is deleted, if the predicted cost $Z(l)+A(l)$ of the whole path from s to t is dominated by the predicted cost $Z(\tilde{l}) + A(\tilde{l})$ of another label $\tilde{l}$ at the same node with the same prediction matrix.
Using the dominance relation [$R^{\text{point}}/R^{\text{set}}$], the resulting algorithm [EL-PB/EL-SB] finds a complete set of [pointMR/setMR] efficient solutions for conservative instances (Publication 2, Theorem 16).

The results of this section are summarized in Table 3.1. All presented algorithms run in pseudo-polynomial time for integer edge costs, if the number of objectives and scenarios is fixed.

Experiments

The authors test their algorithms on two types of networks. *Grid networks* contain a set of nodes that can be interpreted as a two-dimensional grid where vertically or horizontally neighbored nodes are connected by edges. An additional [start/termination]

Concept of robust efficiency	Cond. 1	Cond. 2	Algorithms
multi-scenario efficiency	yes	yes	Algorithm 1
flimsily robust efficiency	yes	no	EL-Flimsily, RL-Flimsily
highly robust efficiency	yes	no	EL-Highly, RL-Highly
pointMR efficiency	no	yes	EL-PB
setMR efficiency	no	yes	EL-SB

Table 3.1.: Summary of which conditions are satisfied for which concept of robust efficiency and which algorithms can be used to find a complete set of robust efficient solutions (Publication 2, Table 1).

node is connected to all nodes in the [first/last] column of the grid. In *NetMaker networks* the nodes are enumerated such that s is the first and t the last node. Most edges connect nodes whose indices do not differ more than a given threshold. This prevents paths from s to t with very few edges, which would then easily dominate all other paths.
To obtain so called *random instances*, the integer edge costs are generated randomly from given intervals, for details see Addendum A.2, pages 83–84. To obtain instances with correlated scenarios (called *correlated instances*), only the cost matrices for the first scenario are generated in this way and the costs for all other scenarios are chosen similar to the first scenario.

The main focus of the experimental evaluation is to compare the extended and the repeated labeling algorithms. Therefore, the authors mainly investigate the performance of the algorithms EL-Flimsily and RL-Flimsily (as the runtimes of the corresponding algorithms for finding highly robust efficient solutions are similar). We summarize the main results.

- The repeated labeling algorithm RL-Flimsily solves most instances faster than the extended labeling algorithm EL-Flimsily. This can be explained by the high number of labels created and kept at each node in EL-Flimsily, because a label is only deleted when it has been detected to be dominated in every scenario.

- The runtime of both algorithms mostly increases with the number of objectives, the number of scenarios and the size of the network. In case of grid networks, the width plays a more important role than the height.

- The performance of RL-Flimsily is similar for random and correlated instances, whereas EL-Flimsily solves correlated instances much faster than random instances. An explanation for this is that, for correlated scenarios, a label dominating another in one scenario tends to dominate it in all other scenarios as well, hence for discarding a label in EL-Flimsily, often one dominating label suffices. RL-Flimsily, on the other hand, executes Algorithm 1' once for each scenario, even if the costs are identical for all scenarios. For some correlated instances, in particular for NetMaker instances with higher numbers of objectives and scenarios, EL-Flimsily is faster than RL-Flimsily.

- For grid networks, all edge cost components for an instance were either chosen from $\{1, ..., 10\}$ or from $\{1, ..., 100\}$. For random instances, the runtime of both algorithms is higher if the wider range is used. For correlated scenarios, however, EL-Flimsily runs faster on instances with the wider cost range, whereas RL-Flimsily runs slower.

- The authors also investigate the number of solutions. Increasing the number of objectives or the network size leads to an increasing number of flimsily and highly robust efficient solutions. For grid networks, the number of flimsily robust efficient solutions increases more with increasing width than with increasing height, which explains the different influence of the grid's width and height on the runtime of the algorithms. When the number of scenarios increases, the number of flimsily robust efficient solutions tends to increase, whereas the number of highly robust efficient solutions tends to decrease. Correlated instances have more highly robust efficient solutions than random instances.

In addition, the authors investigate the performance of EL-PB and EL-SB on small grid network instances. Besides the network size and the number of objectives and scenarios, the values of the edge cost components strongly influence the performance of both algorithms as well: the runtime increases significantly for an increasing number of prediction matrices, which is determined by the edge cost components. The runtime of EL-PB is lower and increases more slowly than that of EL-SB, because the dominance check in EL-PB takes less time. In comparison, the algorithms for finding flimsily and highly robust efficient solutions are much faster and can be used for considerably bigger instances.

Own Contribution

This paper is joint work with A. Raith, M. Schmidt and A. Schöbel. Even though the ideas were developed and discussed among all authors, most of the ideas for the algorithms were introduced by myself. Also, the major part of the details, including algorithms and examples, were my contribution. I have written most of the theoretical chapter and a smaller part of the experimental evaluation. The main part of the implementations was done by myself and a student I supervised. I have done part of the experiments and their analysis, though the major part of the experiments was conducted and evaluated by A. Raith.

3.3. Min-Ordering and Max-Ordering Scalarization Methods for Multi-Objective Robust Optimization

In Schmidt et al. (2018), referred to as Publication 3, two scalarization methods for multi-objective uncertain optimization problems are developed: the min-ordering and the max-ordering method. The authors show that all pointMR weakly efficient solutions can be found with the max-ordering method. The min-ordering method finds setMR weakly efficient solutions, some of which cannot be found with previously known scalarization methods. The authors investigate how to approach the resulting scalarized problems for combinatorial optimization problems with particular uncertainty sets. In case of interval uncertainty they show that the uncertainty set can be reduced to one scenario. They introduce two versions of bounded uncertainty and develop compact mixed integer linear programming (MILP) formulations for the scalarized problems with these uncertainty sets.

Min-Ordering and Max-Ordering Scalarization Methods for Multi-Objective Uncertain Problems

Given a multi-objective uncertain optimization problem (MOUP) as in Definition 2.7, a weight vector $\lambda \in \mathbb{R}^k_>$ and a reference point $r \in \mathbb{R}^k$, the authors of Publication 3 define the corresponding *min-ordering optimization problem* as

$$\text{P-min}(r,\lambda) \quad \min_{x\in\mathcal{X}} \alpha^{\min}(x,r,\lambda) \qquad \text{with } \alpha^{\min}(x,r,\lambda) := \max_{\xi\in\mathcal{U}} \min_{i\in\{1,\dots,k\}} \lambda_i \left(z_i(x,\xi) - r_i\right)$$

and the corresponding *max-ordering optimization problem* as

$$\text{P-max}(r,\lambda) \quad \min_{x\in\mathcal{X}} \alpha^{\max}(x,r,\lambda) \qquad \text{with } \alpha^{\max}(x,r,\lambda) := \max_{\xi\in\mathcal{U}} \max_{i\in\{1,\dots,k\}} \lambda_i \left(z_i(x,\xi) - r_i\right).$$

The *min-ordering* (resp. *max-ordering*) *scalarization method* is obtained, similar to other scalarization methods, by varying the parameters r and λ and solving the resulting problems P-min(r,λ) (resp. P-max(r,λ)).
The authors remark that the max-ordering scalarization method is similar to the weighted Chebyshev method (Ide, 2014; Hassanzadeh et al., 2013), but with arbitrary reference point. They further remark that for $|\mathcal{U}| = 1$, the min-ordering scalarization problem can be solved by solving k single-objective deterministic problems, whereas the max-ordering problem is then equivalent to a single-objective minmax robust optimization problem with k scenarios.

Geometric Characterization

The authors provide a geometric interpretation of the solutions for P-max(r, λ) and P-min(r, λ). With $g(r, \lambda)$ being the line

$$g(r, \lambda) := \left\{ r + \alpha \left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k} \right)^T : \alpha \in \mathbb{R} \right\},$$

and ∂M denoting the boundary of a set $M \subseteq \mathbb{R}^k$, they give the following characterization.

Theorem 3.2 (Publication 3, Theorem 10)**.** *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ be given. A feasible solution $x^* \in \mathcal{X}$ is optimal for P-max(r, λ) if and only if there exists $y^* \in \mathbb{R}^k$ such that (x^*, y^*) is an efficient solution for*

$$\begin{array}{llrcl} \text{G-max}(r,\lambda) & & \min & y & \\ & & s.t. & y & \in & g(r, \lambda) \cap \partial(\bar{z}(x) + \mathbb{R}^k_{\geqq}) \\ & & & x & \in & \mathcal{X}. \end{array}$$

A feasible solution $x^* \in \mathcal{X}$ is optimal for P-min(r, λ) if and only if there exists $y^* \in \mathbb{R}^k$ such that (x^*, y^*) is an efficient solution for

$$\begin{array}{llrcl} \text{G-min}(r,\lambda) & & \min & y & \\ & & s.t. & y & \in & g(r, \lambda) \cap \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}) \\ & & & x & \in & \mathcal{X}. \end{array}$$

This means that the optimal solutions for P-max(r, λ) can be identified by comparing the intersection points of $g(r, \lambda)$ with $\partial(\bar{z}(x) + \mathbb{R}^k_{\geqq})$ for all $x \in \mathcal{X}$. Similarly, the optimal solutions of P-min(r, λ) can be identified by comparing the intersection points of $g(r, \lambda)$ with $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$ for all $x \in \mathcal{X}$. Figure 3.2 shows $\partial(\bar{z}(x) + \mathbb{R}^k_{\geqq})$, $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$ and $g(r, \lambda)$ for an example (Publication 3, Example 6). With help of the intersection points, it is easy to see that for $r = (0, 1)^T, \lambda = (3, 4)^T$, x^1 is uniquely optimal for P-max(r, λ) and x^2 is uniquely optimal for P-min(r, λ).
In addition, Figure 3.2(b) shows the set $\tilde{Y}$ of all points in $Y := \bigcup_{x \in \mathcal{X}} \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$, such that no $y' \in Y$ with $y' < y$ exists. For every point in $\tilde{Y}$, there exist $x \in \mathcal{X}$, $r \in \mathbb{R}^k$, $\lambda \in \mathbb{R}^k_>$ such that (x, y) is an efficient solution for G-min(r, λ).

Minmax Robust Efficient Solutions Found with the Min-Ordering and Max-Ordering Scalarization Methods

The authors show that optimal solutions for P-max(r, λ) are pointMR weakly efficient for MOUP and that every pointMR weakly efficient solution for MOUP can be found with the max-ordering scalarization method.

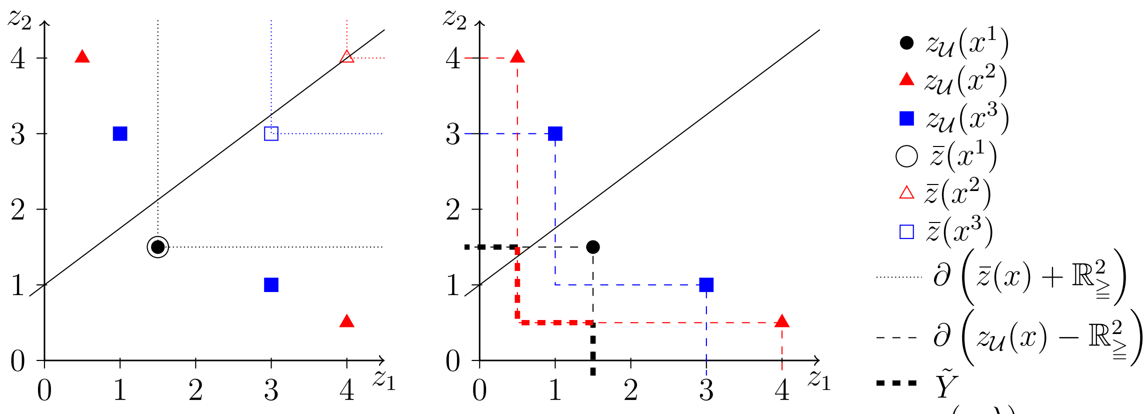


Figure 3.2.: Determining the intersection point of $g(r, \lambda)$ with $\partial(\bar{z}(x) + \mathbb{R}^2_{\geqq})$ in (a) and $\partial(z_\mathcal{U}(x) - \mathbb{R}^2_{\geqq})$ in (b); as an example, $g(r, \lambda)$ is shown for $r = (0, 1)^T$, $\lambda = (3, 4)^T$ (Publication 3, Figure 3).

Theorem 3.3 (Publication 3, Theorem 11)**.** *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ be given and let x be an optimal solution for P-max(r, λ). Then*

1. *x is a pointMR weakly efficient solution for MOUP and*
2. *if x is the unique optimal solution for P-max(r, λ), then x is a pointMR strictly efficient solution for MOUP.*

Theorem 3.4 (Publication 3, Theorem 12)**.** *Let x be a pointMR weakly efficient solution for MOUP and let a reference point $r \in \mathbb{R}^k$ with $r_i < \max_{\xi \in \mathcal{U}} z_i(x, \xi)$ $\forall i \in \{1, ..., k\}$ be given. Then there exists a weight vector $\lambda \in \mathbb{R}^k_>$ such that x is an optimal solution for P-max(r, λ).*

The min-ordering scalarization method, on the other hand, finds setMR weakly efficient solutions.

Theorem 3.5 (Publication 3, Theorem 13)**.** *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ be given and let x be an optimal solution for P-min(r, λ). Then*

1. *x is a setMR weakly efficient solution for MOUP and*
2. *if x is the unique optimal solution for P-min(r, λ), then x is a setMR strictly efficient solution for MOUP.*

Moreover, the authors show that optimal solutions for P-min(r, λ) cannot necessarily be found with any of the other known scalarization methods for finding setMR (weakly) efficient solutions, namely the weighted sum, ϵ-constraint, (augmented)

weighted Chebychev, p-norm or max-ordering scalarization method (Publication 3, Theorem 14). That is, with help of the min-ordering optimization problem one can find "new" setMR efficient solutions. In the example shown in Figure 3.2, the solution x^2 is optimal for P-min(r, λ), but only x^1 is found with the other methods, whereas x^3 is neither found with the min-ordering method nor with any of the other scalarization methods.
The authors remark that even though P-min(r, λ) could be easily transformed to a (single-objective) adjustable robust problem, a potential structure of the underlying problem (e.g., being a linear problem or a particular combinatorial problem) would often be lost.

Min-Ordering and Max-Ordering Problems for MOUCO with Interval Uncertainty

Furthermore, the authors of Publication 3 investigate how to approach P-min(r, λ) and P-max(r, λ) for multi-objective uncertain combinatorial optimization problems with particular uncertainty sets. They represent the solutions of a combinatorial optimization problem as binary vectors $x \in \{0, 1\}^n$, as explained in Section 2.3.
First, they consider the straightforward multi-objective extension of interval uncertainty

$$\mathcal{U}^I := \left\{ c \in \mathbb{R}^{k \times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0, 1] \; \forall i \in \{1, ..., k\}, j \in \{1, ..., n\} \right\},$$

with given minimal values $\hat{c}_{i,j} \in \mathbb{R}$ and interval lengths $\delta_{i,j} \in \mathbb{R}_{\geqq}$. They show that instead of considering $\mathcal{U}^I$, it is sufficient to solve the problem with an uncertainty set only containing one scenario $\bar{c}$, defined by $\bar{c}_{i,j} := \hat{c}_{i,j} + \delta_{i,j}$. It follows that P-min$(r, \lambda)$ can be solved in polynomial time, if the underlying single-objective deterministic problems $\min_{x \in \mathcal{X}} z_i(x, c)$ are polynomially solvable, while P-max(r, λ) is NP-hard for many combinatorial problems (e.g., the shortest path or minimum spanning tree problem).

Min-Ordering and Max-Ordering Problems for MOUCO with Bounded Uncertainty

The authors introduce an extension of bounded uncertainty (Definition 2.6) to the multi-objective case, where the number of all uncertain parameters deviating from their nominal value is bounded (instead of regarding each objective independently as in $\mathcal{U}^{\text{owb}}$ in Definition 2.15). As they show later, in contrast to the problems and uncertainty sets considered in Bertsimas and Sim (2003); Hassanzadeh et al. (2013); Raith et al. (2018b), for P-min(r, λ) they need to distinguish between integer and real valued factors $\beta_{i,j}$.

Definition 3.6 (Publication 3, Definitions 18 and 19). *Let* $\hat{c} \in \mathbb{R}^{k \times n}, \delta \in \mathbb{R}^{k \times n}_{\geqq}$ *and* $\Gamma \in \mathbb{Z}$ *with* $0 \leqq \Gamma \leqq (k \cdot n)$ *be given. We define the* discretely bounded uncertainty set *as*

$$\mathcal{U}^d := \Bigg\{ c \in \mathbb{R}^{k\times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in \{0,1\}\ \forall i \in \{1,...,k\}, j \in \{1,...,n\}, \sum_{i\in\{1,\dots,k\}, j\in\{1,\dots,n\}} \beta_{i,j} \leqq \Gamma \Bigg\}$$

and the continuously bounded uncertainty set *as*

$$\mathcal{U}^c := \Bigg\{ c \in \mathbb{R}^{k\times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0,1]\ \forall i \in \{1,...,k\}, j \in \{1,...,n\}, \sum_{i\in\{1,\dots,k\}, j\in\{1,\dots,n\}} \beta_{i,j} \leqq \Gamma \Bigg\}.$$

For these uncertainty sets, the authors of Publication 3 develop compact MILP formulations for P-max(r,λ) and P-min(r,λ), i.e., formulations without any inner maximum or minimum function.

First, the authors show that for P-max(r,λ) one does not need to distinguish between $\mathcal{U}^d$ and $\mathcal{U}^c$, and that both can even be replaced by an objective-wise bounded uncertainty set $\mathcal{U}^{\text{owb}}$ with $\Gamma_1 = \ldots = \Gamma_k$ (Publication 3, Lemma 21).
They conclude that to obtain a compact formulation for P-max(r,λ) with uncertainty set $\mathcal{U}^d$ or $\mathcal{U}^c$, the same approach as in Hassanzadeh et al. (2013) for linear problems with uncertainty set $\mathcal{U}^{\text{owb}}$ can be applied: The authors substitute the inner maximum functions in the objective function by k constraints, which contain embedded optimization problems $\max_{c\in\mathcal{U}} \lambda_i(z_i(x,c) - r_i)$. Then, they replace each of these embedded problems with its dual in the form introduced by Bertsimas and Sim (2003), obtaining a compact MILP formulation for P-max(r,λ).

For P-min(r,λ) with continuously bounded uncertainty set $\mathcal{U}^c$ the authors proceed in a similar way: they find a linear programming formulation of the inner problem $\max_{c\in\mathcal{U}^c} \min_{i\in\{1,\dots,k\}} \lambda_i(z_i(x,c) - r_i)$ and dualize it, using the results by Bertsimas and Sim (2003).
The same approach is not suitable for P-min(r,λ) with discretely bounded uncertainty set $\mathcal{U}^d$, because the inner maximization problem is not a linear program, and is not equivalent to its linear relaxation, which is identical to the inner problem obtained for $\mathcal{U}^c$ (see Publication 3, Example 22). Nevertheless, the authors develop a compact MILP formulation for P-min(r,λ) with help of the identity in Theorem 3.8.

Definition 3.7 (Publication 3, Definitions 23 and 25)**.** *Let δ be a vector in $\mathbb{R}^n$ or a matrix in $\mathbb{R}^{k\times l}$ and let an index set $I \subseteq \{1,\dots,n\}$ resp. $I \subseteq \{1,\dots,k\} \times \{1,\dots,l\}$*

be given. We denote the h-smallest of all entries δ_j with $j \in I$ as $h\text{-}\min_I \delta$ and the h-greatest as $h\text{-}\max_I \delta$. For $h = 0$ or $h > |I|$ we set $h\text{-}\min_I \delta = h\text{-}\max_I \delta = 0$.
Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ and $x \in \mathcal{X}$ be given. We define $M(x) \in \mathbb{R}^{k\times(\Gamma+1)}$ by its entries

$$m_{i,l} := \lambda_i \left(-r_i + \sum_{j\in I(x)} \hat{c}_{i,j} + \sum_{h=1}^{l-1} h\text{-}\max_{I(x)} \delta_{(i,\cdot)} \right),$$

where $I(x) := \{j \in \{1,...,n\} : x_j = 1\}$. That is, $\frac{m_{i,l}+r_i}{\lambda_i}$ is the sum of the nominal cost of x in the i-th objective and the l highest interval lengths $\delta_{i,j}$ among those with $x_j = 1$ w.r.t. the i-th objective.

Theorem 3.8 (Publication 3, Theorem 27)**.** *Given $x \in \mathcal{X}$ and the corresponding matrix $M(x)$, the optimal objective value z^* of the inner maximization problem of P-min(r,λ) equals the $(\Gamma+1)$-smallest entry in $M(x)$, i.e.,*

$$z^* := \max_{c\in\mathcal{U}^d} \min_{i\in\{1,...,k\}} \lambda_i(z_i(x,c) - r_i) = (\Gamma+1)\text{-}\min_{\{1,...,k\}\times\{1,...,\Gamma+1\}} M(x) =: m^*.$$

The authors formulate a compact mixed integer linear program that minimizes the $(\Gamma+1)$-smallest entry in $M(x)$ over all $x \in \mathcal{X}$, which hence is a compact MILP formulation for P-min(r,λ) with discretely bounded uncertainty set.

Further, the authors show that the complexity of P-min(r,λ) and P-max(r,λ) with bounded uncertainty depends on Γ. For $\Gamma = 0$ the uncertainty set only contains one scenario and P-max(r,λ) is already NP-hard for many combinatorial problems, as the shortest path and minimum spanning tree problem, whereas P-min(r,λ) is polynomially solvable as long as the underlying deterministic problem is polynomially solvable. However, for $\Gamma = 1$, P-min(r,λ) with discretely bounded uncertainty set is NP-hard for the shortest path and minimum spanning tree problem, which is shown by reducing the single-objective minmax robust problem with discrete uncertainty set to P-min(r,λ) (Publication 3, Theorem 29).

Own contribution

This manuscript is joint work with M. Schmidt and A. Schöbel. The initial idea for the scalarizations was contributed by M. Schmidt, but the ideas for most theorems and solution approaches were elaborated jointly by the authors. The idea for the geometrical characterization and the proof that P-min(r,λ) finds "new" setMR efficient solutions is my own work. A big part of the details, including most of the proofs, models, examples and figures, as well as the major part of the writing, were done by myself with consultation of the other authors.

4. Discussion

Combinatorial problems have been extensively studied in multi-objective optimization and robust optimization, yet hardly any approaches to find robust efficient solutions for multi-objective uncertain combinatorial problems have existed previous to the work presented in this thesis. The field of multi-objective robust optimization is currently gaining more and more interest, as many of the concepts of robust efficiency have been developed during the last years. Moreover, there is an inherent difficulty to multi-objective robust optimization, as it combines challenges arising from both the uncertainties and the multi-objective nature of the problems.
This thesis contributes to the analysis of robust efficient solutions and the development of solution methods for multi-objective uncertain combinatorial problems, considering various types of uncertainty and concepts of robust efficiency. Approaches from both robust optimization and multi-objective optimization are extended and combined, using properties of particular robustness concepts and uncertainty sets.

Beyond solution approaches for general combinatorial problems (see Publications 1 and 3) the shortest path problem is considered in particular (Publications 1 and 2). The shortest path problem is an example for an "easy" problem becoming "hard" in multi-objective robust optimization: it is polynomially solvable in the single-objective deterministic case, but its multi-objective counterpart can have exponentially many nondominated objective vectors and it is, e.g., NP-hard in the minmax robust case with finite uncertainty.
Among the various existing concepts for robust efficiency, a focus is put on pointMR and setMR efficiency, which both extend the popular single-objective concept of minmax robustness. These concepts are considered in all three publications constituting the cumulative part of this thesis. The authors of Publication 1 use the concept of pointMR efficiency, and, since the uncertainty set in this setting is an objective-wise uncertainty set, all pointMR efficient solutions are also setMR efficient and vice versa (see Lemma 2.13). With the scalarization methods introduced in Publication 3, setMR respective pointMR efficient solutions are found. In Publication 1, too, both concepts are considered, as well as flimsily and highly robust efficiency and multi-scenario efficiency.
Furthermore, different uncertainty sets are regarded. Finite uncertainty (considered in Publication 2) and interval uncertainty (considered in Publication 3) are straightforward extensions of the respective uncertainty sets in single-objective robust optimization. The single-objective concept of bounded uncertainty can be extended to

multi-objective robust optimization in several ways. In Publication 1 an objective-wise variant is considered, and the authors of Publication 3 introduce the continuously and discretely bounded uncertainty sets.
Table 4.1 gives an overview of the considered problems, robustness concepts and uncertainty sets in the cumulative part of this thesis. For practical applications with any

Publication	Problems	Concepts	Uncertainty Sets
Publication 1	CO, SP	pointMR, (setMR)	objective-wise bounded
Publication 2	SP	pointMR, setMR, flimsily, highly, multi-scenario	finite
Publication 3	G, CO	pointMR, setMR	discretely/continuously bounded, interval

Table 4.1.: Considered problem classes (G: general optimization problems, CO: general combinatorial problems, SP: the shortest path problem), concepts of robust efficiency and uncertainty sets.

of the combinations of problem type, robustness concept and uncertainty set listed in Table 4.1, a decision maker can now choose an appropriate solution method from the publications summarized in this thesis. In case of several different approaches for the same combination, the authors compare their developed algorithms experimentally and recommend one or another approach depending on the parameters of the problem (see Publications 1 and 2).

Moreover, this thesis indicates how to approach further multi-objective robust uncertain combinatorial optimization problems by showing strategies to find new solution methods based on results from different fields.
On the one hand, algorithms for finding minmax robust solutions for single-objective uncertain problems are analyzed and extended to find pointMR and setMR efficient solutions in Publications 1 and 2.
On the other hand, solution methods based on ideas from multi-objective optimization are developed. In Publication 3, the authors introduce two scalarization methods to obtain pointMR or setMR efficient solutions by solving several scalarized problems, which is a common approach in multi-objective optimization. In Publications 1 and 2, labeling algorithms for the multi-objective deterministic shortest path problem are extended to the multi-objective robust case, sometimes combined with algorithms or solution ideas from robust optimization.
Furthermore, the structure of particular uncertainty sets is used to formulate mixed integer linear programming models for the scalarized problems in Publication 3.
This shows that the combination of uncertainties and multiple objectives leads not

only to very challenging combinatorial problems, but also offers a variety of possible solution approaches.

An advantage of adapting algorithms from robust optimization or multi-objective optimization is that theoretical complexity results can often be retained or derived: For example, the multi-objective labeling algorithm extended in Publication 2 runs in pseudo-polynomial time under some assumptions, as do all presented extensions. Further, the runtime of the algorithms repeatedly applying an algorithm for multi-objective deterministic problems (see Publications 1 and 2) is bounded by a multiple of that algorithm's runtime (plus the time for filtering the solutions, if necessary). This also means that these algorithms directly profit from any advances in the field of multi-objective optimization. The approaches in Publication 1 have the additional advantage that they are valid for all combinatorial problems, and that the subproblems are of the same kind as the original problem, hence specific algorithms for specific combinatorial problems can be applied. On the other hand, the scalarization approach given in Publication 3 can be used for all uncertainty sets, as long as a suitable method for solving the scalarized problems is found.

Furthermore, the solution methods presented in this thesis show that similar approaches are suitable for different robustness concepts and uncertainty sets. For example, labeling algorithms have been used in several of the methods for the shortest path problem, either by extending them or using them to solve auxiliary problems. The scalarization methods introduced in Publication 3 can also be applied to the problems in Publications 1 and 2: The MILP formulation for P-max(r, λ) with discretely and continuously bounded uncertainty is also valid for objective-wise bounded uncertainty as considered in Publication 1, if the Γ_i are identical. Compact MILP formulations can be obtained for P-min(r, λ) and P-max(r, λ) with a finite uncertainty set, too, and can hence be used to find pointMR and setMR efficient solutions for the shortest path problem in Publication 2.

In conclusion, different uncertainty sets and robustness concepts require different solution methods, but often similar approaches can be used to find these methods. Therefore, approaches introduced in this thesis can likely be of help when developing solution methods for problems with other robustness concepts and uncertainty sets. These approaches comprise the extension of algorithms from robust and multi-objective optimization and mixed integer programming formulations. A general superiority of one approach over the others is not observed.

5. Conclusion and Future Work

In this thesis, several solution methods for multi-objective robust combinatorial optimization problems have been developed. In Section 3.1, two approaches for finding pointMR (resp. setMR) efficient solutions for multi-objective combinatorial problems with objective-wise bounded uncertainty have been introduced and applied to the shortest path problem. In Section 3.2, we presented labeling algorithms for the multi-objective shortest path problem with finite uncertainty, finding robust efficient solutions according to the concepts of multi-scenario efficiency, flimsily and highly robust efficiency and pointMR and setMR efficiency. Two scalarization methods for finding pointMR and setMR efficient solutions were introduced in Section 3.3, and approaches to solve the scalarized problems were shown for interval uncertainty and discretely and continuously bounded uncertainty.

All proposed solution methods provide specific issues for future work. For example, it could be worthwhile to investigate whether acceleration methods for labeling algorithms can be applied to the algorithms presented in Sections 3.1 and 3.2, or to combine the scalarization methods from Section 3.3 by using ordered median functions as scalarizing functions.

Furthermore, a great variety of robustness concepts for multi-objective optimization exists, as indicated in Section 2.3. We focused on pointMR and setMR efficiency, and also regarded flimsily and highly robust efficiency and multi-scenario efficiency. Considering different robustness concepts leads to different problems to solve and hence requires different solution methods, as shown for the shortest path problem in Section 3.2. Therefore, one aspect of future work should be to analyze whether solution methods presented in this thesis can be adapted to further concepts of robust efficiency or whether new approaches are needed.

Similarly, the assumed uncertainty set plays an important role for the structure and complexity of the resulting robust problem. We have considered general combinatorial problems with interval uncertainty and several versions of bounded uncertainty, and shortest path problems with objective-wise bounded and finite uncertainty. An interesting research question would be whether the proposed solution methods can be used or adapted for other uncertainty sets, including ellipsoidal, polyhedral and other variants of bounded uncertainty.

The methods presented in Chapter 3 are based on approaches from multi-objective optimization, robust optimization and integer programming. We are far from exhausting the results and methods from these long-established fields, leaving much opportunity for further solution approaches for multi-objective robust combinatorial

problems.
Apart from developing methods for general combinatorial problems, we have shown how to extend specific algorithms for the (robust or multi-objective) shortest path problem to the multi-objective robust case. It is of interest whether other solution methods for the shortest path problem could be extended as well. Specific algorithms for other combinatorial problems, e.g., the multi-objective robust minimum spanning tree problem, are yet to be developed.
In conclusion, covering several robustness concepts and uncertainty sets and providing a variety of solution methods, this thesis gives an insight into the challenges of multi-objective robust combinatorial optimization and how to approach them. It is one of the first contributions on the way to an extensive analysis of and solution concept for multi-objective robust combinatorial optimization. As emphasized in the introduction and the application sections, this is also of practical relevance, since real-world optimization problems are often of a multi-objective and uncertain nature.

Bibliography

Avigad, G. and Branke, J. (2008). Embedded evolutionary multi-objective optimization for worst case robustness. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*, pages 617–624. ACM, New York.

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press, Princeton and Oxford.

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

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71.

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

Bitran, G. R. (1980). Linear multiple objective problems with interval coefficients. *Management Science*, 26(7):694–706.

Bokrantz, R. and Fredriksson, A. (2017). Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. *European Journal of Operational Research*, 262(2):682–692.

Botte, M. and Schöbel, A. (2016). Dominance for multi-objective robust optimization. Technical Report 2016-8, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen.

Bowman, V. J. (1976). On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In Thiriez, H. and Zionts, S., editors, *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and*

Mathematical Systems (Operations Research), pages 76–86. Springer, Berlin, Heidelberg.

Büsing, C. and D'Andreagiovanni, F. (2014). A new theoretical framework for robust optimization under multi-band uncertainty. In Helber, S., Breitner, M., Rösch, D., Schön, C., Graf von der Schulenburg, J.-M., Sibbertsen, P., Steinbach, M., Weber, S., and Wolter, A., editors, *Operations Research Proceedings 2012*, pages 115–121. Springer, Cham.

Chankong, V. and Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*. Number 8 in North Holland Series in System Science and Engineering. Elsevier, New York.

Chassein, A., Dokka, T., and Goerigk, M. (2018). Algorithms and uncertainty sets for data-driven robust shortest path problems. *ArXiv e-prints*. https://arxiv.org/abs/1802.04149v1.

Cicerone, S., D'Angelo, G., Di Stefano, G., Frigioni, D., and Navarra, A. (2007). Robust algorithms and price of robustness in shunting problems. In Liebchen, C., Ahuja, R. K., and Mesa, J. A., editors, *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS07)*, volume 7 of *OpenAccess Series in Informatics (OASIcs)*, pages 175–190. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.

Cintrano, C., Chicano, F., and Alba, E. (2017). Robust bi-objective shortest path problem in real road networks. In Alba, E., Chicano, F., and Luque, G., editors, *Smart Cities*, pages 128–136. Springer, Cham.

Corley, H. and Moon, I. (1985). Shortest paths in networks with vector weights. *Journal of Optimization Theory and Applications*, 46(1):79–86.

Deb, K. and Gupta, H. (2006). Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Doolittle, E., Kerivin, H. M., and Wiecek, M. M. (2012). A robust multiobjective optimization problem with application to internet routing. Technical Report R2012-11-DKW, Clemson University.

Doolittle, E. K., Dranichak, G. M., Muir, K., and Wiecek, M. M. (2016). A note on robustness of the min-max solution to multi-objective linear programs. *International Journal of Multicriteria Decision Making*, 6(4):343–365.

Edgeworth, F. Y. (1881). *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. C. Kegan Paul & Co., London.

Ehrgott, M. (2005). *Multicriteria optimization*. Springer, Berlin, Heidelberg.

Ehrgott, M. (2006). A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360.

Ehrgott, M. and Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460.

Ehrgott, M., Ide, J., and Schöbel, A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31.

Eichfelder, G., Krüger, C., and Schöbel, A. (2017). Decision uncertainty in multiobjective optimization. *Journal of Global Optimization*, 69(2):485–510.

Erera, A. L., Morales, J. C., and Savelsbergh, M. (2009). Robust optimization for empty repositioning problems. *Operations Research*, 57(2):468–483.

Fischetti, M. and Monaci, M. (2009). Light robustness. In Ahuja, R. K., Möhring, R. H., and Zaroliagis, C. D., editors, *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, pages 61–84. Springer, Berlin, Heidelberg.

Fliege, J. and Werner, R. (2014). Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research*, 234(2):422–433.

Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics*, 2(1-2):39–45.

Goerigk, M. and Schöbel, A. (2016). Algorithm engineering in robust optimization. In Kliemann, L. and Sanders, P., editors, *Algorithm Engineering: Selected Results and Surveys*, volume 9220 of *Lecture Notes in Computer Science*, pages 245–279. Springer, Cham.

Guerriero, F. and Musmanno, R. (2001). Label correcting methods to solve multicriteria shortest path problems. *Journal of Optimization Theory and Applications*, 111(3):589–613.

Gunawan, S. and Azarm, S. (2005). Multi-objective robust optimization using a sensitivity region concept. *Structural and Multidisciplinary Optimization*, 29(1):50–60.

Haimes, Y. Y., Lasdon, L. S., and Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297.

Hamacher, H. W. and Ruhe, G. (1994). On spanning tree problems with multiple objectives. *Annals of Operations Research*, 52(4):209–230.

Hansen, P. (1980). Bicriterion path problems. In Fandel, G. and Gal, T., editors, *Multiple Criteria Decision Making Theory and Application*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 109–127. Springer, Berlin, Heidelberg.

Hassanzadeh, F., Nemati, H., and Sun, M. (2013). Robust optimization for multi-objective programming problems with imprecise information. *Procedia Computer Science*, 17:357–364.

Hombach, L. E., Büsing, C., and Walther, G. (2017). Robust and sustainable supply chains under market uncertainties and different risk attitudes—A case study of the german biodiesel market. *European Journal of Operational Research*.

Iancu, D. A. and Trichakis, N. (2014). Pareto efficiency in robust optimization. *Management Science*, 60(1):130–147.

Ide, J. (2014). *Concepts of Robustness for Uncertain Multi-Objective Optimization*. PhD thesis, Universität Göttingen.

Ide, J., Köbis, E., Kuroiwa, D., Schöbel, A., and Tammer, C. (2014). The relationship between multi-objective robustness concepts and set-valued optimization. *Fixed Point Theory and Applications*, 2014(83).

Ide, J. and Schöbel, A. (2016). Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271.

Ide, J., Tiedemann, M., Westphal, S., and Haiduk, F. (2015). An application of deterministic and robust optimization in the wood cutting industry. *4OR*, 13(1):35–57.

Iori, M., Martello, S., and Pretolani, D. (2010). An aggregate label setting policy for the multi-objective shortest path problem. *European Journal of Operational Research*, 207(3):1489–1496.

Kasperski, A. and Zieliński, P. (2016). Robust discrete optimization under discrete and interval uncertainty: A survey. In *Robustness Analysis in Decision Aiding, Optimization, and Analytics*, pages 113–143. Springer, Cham.

Klamroth, K., Köbis, E., Schöbel, A., and Tammer, C. (2017). A unified approach to uncertain optimization. *European Journal of Operational Research*, 260(2):403–420.

Kouvelis, P. and Yu, G. (1997). *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston.

Krüger, C. (2018a). *On Minmax Robustness for Multiobjective Optimization with Decision or Parameter Uncertainty*. PhD thesis, Universität Göttingen.

Krüger, C. (2018b). Peat and pots: Analysis of robust solutions for a biobjective problem in agriculture. Technical Report 2018-5, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen.

Krüger, C., Castellani, F., Geldermann, J., and Schöbel, A. (2018). Peat and pots: An application of robust multiobjective optimization to a mixing problem in agriculture. Technical Report 2018-4, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen.

Krüger, C., Schöbel, A., and Wiecek, M. M. (2017). The robustness gap for uncertain multiobjective optimization. Technical Report 2017-3, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen.

Kuhn, K., Raith, A., Schmidt, M., and Schöbel, A. (2016). Bi-objective robust optimisation. *European Journal of Operational Research*, 252(2):418–431.

Kuroiwa, D. and Lee, G. M. (2012). On robust multiobjective optimization. *Vietnam Journal of Mathematics*, 40(2-3):305–317.

Lee, T. and Kwon, C. (2014). A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR*, 12(4):373–378.

Liebchen, C., Lübbecke, M., Möhring, R., and Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In Ahuja, R., Möhring, R., and Zaroliagis, C., editors, *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, volume 5868 of *Lecture Notes in Computer Science*, pages 1–27. Springer, Berlin, Heidelberg.

Majewski, D. E., Wirtz, M., Lampe, M., and Bardow, A. (2017). Robust multiobjective optimization for sustainable design of distributed energy supply systems. *Computers & Chemical Engineering*, 102:26–39.

Martins, E. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245.

Mavrotas, G., Figueira, J. R., and Siskos, E. (2015). Robustness analysis methodology for multi-objective combinatorial optimization problems and application to project selection. *Omega*, 52:142 – 155.

Murthy, I. and Her, S. (1992). Solving min-max shortest-path problems on a network. *Naval Research Logistics*, 39(5):669–683.

Nikulin, Y., Karelkina, O., and Mäkelä, M. M. (2013). On accuracy, robustness and tolerances in vector Boolean optimization. *European Journal of Operational Research*, 224(3):449–457.

Paixão, J. M. and Santos, J. L. (2013). Labeling methods for the general case of the multi-objective shortest path problem—A computational study. In Madureira, A., Reis, C., and Marques, V., editors, *Computational Intelligence and Decision Making*, pages 489–502. Springer Netherlands, Dordrecht.

Pareto, V. (1896). *Cours déconomie politique*. F.Rouge, Lausanne.

Park, K.-C. and Lee, K.-S. (2007). A note on robust combinatorial optimization problem. *Management Science and Financial Engineering*, 13(1):115–119.

Poss, M. (2014). Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836–845.

Raith, A., Schmidt, M., Schöbel, A., and Thom, L. (2018a). Extensions of labeling algorithms for multi-objective uncertain shortest path problems. *Networks*.

Raith, A., Schmidt, M., Schöbel, A., and Thom, L. (2018b). Multi-objective min-max robust combinatorial optimization with cardinality-constrained uncertainty. *European Journal of Operational Research*, 267(2):628–642.

Schmidt, M., Schöbel, A., and Thom, L. (2018). Min-ordering and max-ordering scalarization methods for multi-objective robust optimization. Technical Report 2018-3, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen.

Schöbel, A. (2014). Generalized light robustness and the trade-off between robustness and nominal quality. *Mathematical Methods of Operations Research*, 80(2):161–191.

Soyster, A. L. (1973). Technical note—Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157.

Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344.

Wiecek, M. M., Blouin, V. Y., Fadel, G. M., Engau, A., Hunt, B. J., and Singh, V. (2009). Multi-scenario multi-objective optimization with applications in engineering design. In Barichard, V., Ehrgott, M., Gandibleux, X., and T'Kindt, V., editors, *Multiobjective Programming and Goal Programming*, volume 618 of *Lecture Notes in Economics and Mathematical Systems*, pages 283–298. Springer, Berlin, Heidelberg.

Wiecek, M. M. and Dranichak, G. M. (2016). Robust multiobjective optimization for decision making under uncertainty and conflict. In Gupta, A. and Capponi, A., editors, *Optimization Challenges in Complex, Networked and Risky Systems*, INFORMS Tutorials in Operations Research (J.C. Smith, ed.), chapter 4, pages 84–114. INFORMS.

Witting, K., Ober-Blöbaum, S., and Dellnitz, M. (2013). A variational approach to define robustness for parametric multiobjective optimization problems. *Journal of Global Optimization*, 57(2):331–345.

Yu, G. and Yang, J. (1998). On the robust shortest path problem. *Computers & Operations Research*, 25(6):457–468.

Yu, H. and Liu, H. M. (2013). Robust multiple objective game theory. *Journal of Optimization Theory and Applications*, 159(1):272–280.

A. Publications

A.1. Multi-Objective Minmax Robust Combinatorial Optimization with Cardinality-Constrained Uncertainty

published in the *European Journal of Operational Research* (Raith et al., 2018b)

Authors: Andrea Raith, Marie Schmidt, Anita Schöbel, Lisa Thom

European Journal of Operational Research 267 (2018) 628–642

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Multi-objective minmax robust combinatorial optimization with cardinality-constrained uncertainty

Andrea Raith[a], Marie Schmidt[b], Anita Schöbel[c], Lisa Thom[c,*]

[a] Department of Engineering Science, The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand
[b] Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University Rotterdam, PO Box 1738, 3000 DR Rotterdam, The Netherlands
[c] Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestr. 16-18, 37083 Göttingen, Germany

ARTICLE INFO

Article history:
Received 19 December 2016
Accepted 11 December 2017
Available online 18 December 2017

Keywords:
Multiple objective programming
Robust optimization
Combinatorial optimization
Multi-objective robust optimization
Shortest path problem

ABSTRACT

In this paper, we develop two approaches to find minmax robust efficient solutions for multi-objective combinatorial optimization problems with cardinality-constrained uncertainty. First, we extend an existing algorithm for the single-objective problem to multi-objective optimization. We propose also an enhancement to accelerate the algorithm, even for the single-objective case, and we develop a faster version for special multi-objective instances. Second, we introduce a deterministic multi-objective problem with sum and bottleneck functions, which provides a superset of the robust efficient solutions. Based on this, we develop a label setting algorithm to solve the multi-objective uncertain shortest path problem. We compare both approaches on instances of the multi-objective uncertain shortest path problem originating from hazardous material transportation.

1. Introduction

Two of the main difficulties in applying optimization techniques to real-world problems are that several (conflicting) objectives may exist and that parameters may not be known exactly in advance. In multi-objective optimization several objectives are optimized simultaneously by choosing solutions that cannot be improved in one objective without worsening it in another objective. Robust optimization hedges against (all) possible parameter values, e.g., by assuming the worst case for each solution (minmax robustness).

Often it is assumed that the uncertain parameters take any value from a given interval or that discrete scenarios are given. A survey on robust combinatorial optimization with these uncertainty sets is given by Aissi, Bazgan, and Vanderpooten (2009). Based on the interval case, Bertsimas and Sim (2004) propose to consider scenarios where only a bounded number of parameters differ from their expected value (cardinality-constrained uncertainty). This leads to less conservative solutions that are of high practical use. Bertsimas and Sim (2003) provide an algorithm to find robust solutions for combinatorial optimization problems under this kind of uncertainty.

Only recently have robust optimization concepts for multi-objective problems been developed. Kuroiwa and Lee (2012) and Fliege and Werner (2014) introduce a first extension of minmax robustness for several objectives. They consider the uncertainties in the objectives independently of each other. Ehrgott, Ide, and Schöbel (2014) develop another extension of minmax robustness, in which they include the dependencies between the objectives. This is further generalized by Ide, Köbis, Kuroiwa, Schöbel, and Tammer (2014). These concepts have been extensively applied, e.g., in portfolio management (Fliege & Werner, 2014), in game theory (Yu & Liu, 2013) and in the wood industry (Ide, Tiedemann, Westphal, & Haiduk, 2015). Ide and Schöbel (2016) and Wiecek and Dranichak (2016) give an overview on multi-objective robustness, including further robustness concepts. Newest developments in this field include works by Chuong (2016) and Kalantari, Dong, and Davies (2016). Cardinality constrained uncertainty is extended to multi-objective optimization by Doolittle, Kerivin, and Wiecek (2012) (only for uncertain constraints) and Hassanzadeh, Nemati, and Sun (2013) (for uncertain objective functions and constraints).

To the best of our knowledge, only Kuhn, Raith, Schmidt, and Schöbel (2016) have developed a solution algorithm for multi-objective uncertain combinatorial optimization problems. They consider problems with two objectives, of which only one is uncertain, with discrete and polyhedral uncertainty sets.

In this paper, however, we consider problems with any fixed number of objectives of which all may be uncertain. The main

* Corresponding author.
E-mail addresses: a.raith@auckland.ac.nz (A. Raith), schmidt2@rsm.nl (M. Schmidt), schoebel@math.uni-goettingen.de (A. Schöbel), l.thom@math.uni-goettingen.de (L. Thom).

https://doi.org/10.1016/j.ejor.2017.12.018

https://doi.org/10.1016/j.ejor.2017.12.018

A. Raith et al. / European Journal of Operational Research 267 (2018) 628–642

contribution of this paper is that we develop two solution approaches for multi-objective combinatorial optimization problems with cardinality-constrained uncertainty. We further derive specific algorithms for the multi-objective uncertain shortest path problem.

The remainder of this paper is structured as follows: in Section 2 we give a short introduction to multi-objective robust optimization. We present two solution approaches for multi-objective combinatorial optimization problems with cardinality-constrained uncertainty in Section 3: in Section 3.1 we extend an algorithm by Bertsimas and Sim (2003) to multi-objective optimization. Additionally, we propose an acceleration for both the single-objective and the multi-objective case and a faster version for multi-objective problems with a special property. In Section 3.2, we introduce a second approach and show how it can be applied to solve the multi-objective uncertain shortest path problem as an example. In Section 4, we compare our methods on instances of the multi-objective uncertain shortest path problem originating from hazardous material transportation.

2. Multi-objective combinatorial optimization with cardinality-constrained uncertainty

First, we give an introduction to multi-objective combinatorial optimization. We use bold font for vectors and vector valued functions.

An instance $(E, Q, \boldsymbol{c})$ of a multi-objective combinatorial optimization problem is given by a finite element set E, a set $Q \subseteq 2^E$ of feasible solutions, which are subsets of E, and a *cost function* $\boldsymbol{c}$, that assigns a *cost vector* $\boldsymbol{c}(e) = (c_1(e), \dots, c_k(e))$ to each element $e \in E$. The *cost* $\boldsymbol{z}(q)$ of a set $q \in Q$ is the sum of the costs of its elements. We call

$$(MOCO)\ \min_{q\in Q}\left(\boldsymbol{z}(q) = \begin{pmatrix} z_1(e) \\ \vdots \\ z_k(e) \end{pmatrix} = \begin{pmatrix} \sum_{e\in q} c_1(e) \\ \vdots \\ \sum_{e\in q} c_k(e) \end{pmatrix}\right)$$

a *multi-objective combinatorial optimization problem*.

A solution that minimizes all objectives simultaneously does usually not exist. Therefore, we use the well-known concept of *efficient solutions*.

Notation 1. For two vectors $\boldsymbol{y}^1, \boldsymbol{y}^2 \in \mathbb{R}^k$ we use the notation

$$\boldsymbol{y}^1 \leq \boldsymbol{y}^2 \Leftrightarrow y^1_i \leqq y^2_i \text{ for } i = 1, \dots, k \text{ and } \boldsymbol{y}^1 \neq \boldsymbol{y}^2,$$
$$\boldsymbol{y}^1 \leqq \boldsymbol{y}^2 \Leftrightarrow y^1_i \leqq y^2_i \text{ for } i = 1, \dots, k.$$

In the following, we only use the symbols $<$ (strictly less than) and $\leqq$ (less than or equal to) to compare scalars.

Definition 2. A solution $q' \in Q$ *dominates* another solution $q \in Q$ if $\boldsymbol{z}(q') \leq \boldsymbol{z}(q)$. We also say that $\boldsymbol{z}(q')$ dominates $\boldsymbol{z}(q)$. A solution $q \in Q$ is an *efficient* solution, if there is no $q' \in Q$ such that q' dominates q. Then $\boldsymbol{z}(q)$ is called *non-dominated*.

Two efficient solutions $q, q' \in Q$ are called *equivalent* if $\boldsymbol{z}(q) = \boldsymbol{z}(q')$. A set of efficient solutions $\bar{Q} \subseteq Q$ is called *complete* if all $q \in Q \setminus \bar{Q}$ are either dominated by or equivalent to at least one $q' \in \bar{Q}$.

Solving (MOCO) means to find a complete set of efficient solutions.

We now assume that the input data is uncertain, i.e., the feasible set and/or the element costs $\boldsymbol{c}(e)$ are not exactly known in advance. If the set of feasible solutions is uncertain, we aim to find solutions which are feasible in all scenarios (as proposed in the seminal works on robustness, see, e.g., Ben-Tal, El Ghaoui, & Nemirovski, 2009; Soyster, 1973). For this purpose, the sets of feasible solutions can be intersected in advance to obtain a (deterministic) set of *robust feasible solutions*. Hence, in the following, we assume the set Q to be a deterministic set.

The *uncertainty set* $\mathcal{U}$ is then the set of all possible cost functions $\boldsymbol{c}$. The considered uncertainty set often strongly influences the solvability of uncertain optimization problems and the solution approaches. The idea of *cardinality-constrained uncertainty* is to assume that the parameters vary in intervals independent of each other, but not more than a given number of elements will be more expensive than their minimal cost. For example, there will not be an accident on every road of a transportation network at the same time, thus, a delay because of an accident does not need to be considered on all roads simultaneously. Bertsimas and Sim (2003) were the first to introduce cardinality-constrained uncertainty for single-objective uncertain combinatorial optimization problems. With $\hat{c}_e$ being the minimal or *nominal* value of $c(e)$ and $\hat{c}_e + \delta_e$ its maximal value, the considered uncertainty set can be written as

$$\mathcal{U}^{cc} := \{c : E \to \mathbb{R} \mid c(e) \in [\hat{c}_e, \hat{c}_e + \delta_e]\ \forall\, e \in E,$$
$$|\{e \in E \mid c(e) > \hat{c}_e\}| \leqq \Gamma\}. \tag{1}$$

One possible extension to multi-objective optimization is to apply this approach to each objective independently (see Hassanzadeh et al., 2013):

Definition 3. For each element $e \in E$ and each objective i let two real values $\hat{c}_{e,i}$ and $\delta_{e,i} \geqq 0$ be given. We assume that the uncertain cost $c_i(e)$ can take any value in the interval $[\hat{c}_{e,i}, \hat{c}_{e,i} + \delta_{e,i}]$, with $\hat{c}_{e,i}$ being the undisturbed value, called the *nominal* value. For each objective i let an integer $\Gamma_i \leqq |E|$ be given. The *cardinality-constrained uncertainty set* contains all cost functions, with which for each objective i at most Γ_i elements differ from their nominal costs:

$$\mathcal{U}^{mcc} := \{\boldsymbol{c} : E \to \mathbb{R}^k \mid c_i(e) \in [\hat{c}_{e,i}, \hat{c}_{e,i} + \delta_{e,i}]\ \forall\, e \in E,$$
$$\forall\, i = 1, \dots, k, |\{e \in E \mid c_i(e) > \hat{c}_{e,i}\}| \leqq \Gamma_i\ \forall\, i = 1, \dots, k\} \tag{2}$$

We call the family of optimization problems

$$(MOUCO)\ \left(\min_{q\in Q}\left(\boldsymbol{z}(q) = \sum_{e\in q} \boldsymbol{c}(e)\right), \boldsymbol{c} \in \mathcal{U}^{mcc}\right)$$

a *multi-objective uncertain combinatorial optimization problem with cardinality-constrained uncertainty*. An instance of (MOUCO) is hence given by $(E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$, with

$$\hat{C} := \begin{pmatrix} \hat{c}_{e_1,1} & \dots & \hat{c}_{e_1,k} \\ \vdots & & \vdots \\ \hat{c}_{e_{|E|},1} & \dots & \hat{c}_{e_{|E|},k} \end{pmatrix}, \Delta := \begin{pmatrix} \delta_{e_1,1} & \dots & \delta_{e_1,k} \\ \vdots & & \vdots \\ \delta_{e_{|E|},1} & \dots & \delta_{e_{|E|},k} \end{pmatrix},$$
$$\boldsymbol{\Gamma} := (\Gamma_1, \dots, \Gamma_k).$$

Note that with the uncertainty set $\mathcal{U}^{mcc}$, (MOUCO) is *objective-wise uncertain*, as it was defined by Ehrgott et al. (2014), i.e., the uncertainty sets in the objective functions are independent of each other.

This can usually be assumed, if the objectives are uncorrelated. However, also for correlated nominal values, the uncertainty can often be assumed to be uncorrelated, if unexpected events influence only one of the objectives.

To decide what is a good solution for a multi-objective uncertain problem is not trivial. In single-objective robust optimization one looks for so-called *robust optimal* solutions. Often these are defined as solutions, which have a minimal worst case value, i.e., one solves $\min_{q\in Q} \max_{c\in\mathcal{U}} z(q)$ (see, e.g., Ben-Tal et al., 2009). This concept has been generalized to robust efficiency for multi-objective problems in various ways (see, e.g., Ehrgott et al., 2014; Kuroiwa & Lee, 2012). In this paper we determine the worst case independently for each objective (see Definition 4), as proposed by Kuroiwa and Lee (2012). This yields a single vector for each solution and these vectors can be compared using the methods of multi-objective optimization.

https://doi.org/10.1016/j.ejor.2017.12.018

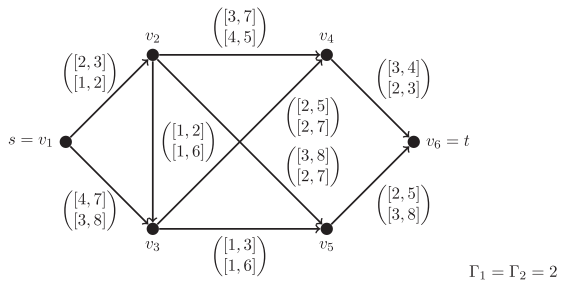


Fig. 1. An instance for (MOUSP).

Definition 4. A solution $q \in Q$ is *robust efficient* for (MOUCO) if q is an efficient solution for the robust counterpart

$$(MORCO) \quad \min_{q \in Q} \left(\boldsymbol{z}^{\mathbf{R}}(q) = \begin{pmatrix} \sup_{\boldsymbol{c} \in \mathcal{U}^{mcc}} z_1(q) \\ \vdots \\ \sup_{\boldsymbol{c} \in \mathcal{U}^{mcc}} z_k(q) \end{pmatrix} = \begin{pmatrix} \sup_{\boldsymbol{c} \in \mathcal{U}^{mcc}} \sum_{e \in q} c_1(e) \\ \vdots \\ \sup_{\boldsymbol{c} \in \mathcal{U}^{mcc}} \sum_{e \in q} c_k(e) \end{pmatrix} \right).$$

Remark 5. Since (MOUCO) is objective-wise uncertain, robust efficiency, as defined in Definition 4, coincides with point-based and set-based minmax robust efficiency defined by Ehrgott et al. (2014). Therefore, all results shown in this paper are valid for both concepts.

Analogously to Definition 2 we define:

Definition 6. Two robust efficient solutions q, $q' \in Q$ are called *equivalent* if $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{z}^{\mathbf{R}}(q')$. A set of robust efficient solutions $\bar{Q} \subseteq Q$ is called *complete* if all $q \in Q \setminus \bar{Q}$ are either dominated w.r.t. $\boldsymbol{z}^{\mathbf{R}}$ or equivalent to at least one $q' \in \bar{Q}$.

2.1. Example: a multi-objective uncertain shortest path problem

Consider a graph $G = (V, E)$ with node set V and edge set E, a start node $s \in V$ and a destination node $t \in V$. A *path* is a sequence of edges connecting adjacent nodes. In a *simple path* at most two edges are incident to each node. For a given cost function $\boldsymbol{c} : E \to \mathbb{R}^k$ the cost of a path is obtained by following the path and adding up the costs of the edges traversed. Because simple paths do not contain any edge more than once, for a simple path q we have $\boldsymbol{z}(q) = \sum_{e \in q} \boldsymbol{c}(e)$.

In the following, we assume *conservative* edge costs, i.e., every cycle C has non-negative cost $z_i(C) \geqq 0$ for each cost function $\boldsymbol{c} \in \mathcal{U}^{mcc}$ and objective $i = 1, \ldots, k$. Then, there always exists a complete set of robust efficient paths containing only simple paths. Hence, the *multi-objective shortest path problem with cardinality-constrained uncertainty* can be written as a combinatorial problem

$$(MOUSP) \quad \left(\min_{q \in Q} \sum_{e \in q} \boldsymbol{c}(e), \boldsymbol{c} \in \mathcal{U}^{mcc} \right)$$

with Q being the set of simple paths from s to t in G. We use the following example to illustrate the results and algorithms in this paper.

Example 7. Consider the network in Fig. 1 with $s = v_1$ and $t = v_6$ and $\Gamma_1 = \Gamma_2 = 2$. The edge costs are given in the form

$$\begin{pmatrix} [\hat{c}_{e,1}, \hat{c}_{e,1} + \delta_{e,1}] \\ [\hat{c}_{e,2}, \hat{c}_{e,2} + \delta_{e,2}] \end{pmatrix}.$$

For this instance of (MOUSP), the set of robust efficient paths consists of the two paths

$$q_1 := \{(v_1, v_2), (v_2, v_4), (v_4, v_6)\} \text{ with } \boldsymbol{z}^{\mathbf{R}}(q_1) = \begin{pmatrix} 13 \\ 9 \end{pmatrix},$$

$$q_2 := \{(v_1, v_2), (v_2, v_3), (v_3, v_5), (v_5, v_6)\} \text{ with } \boldsymbol{z}^{\mathbf{R}}(q_2) = \begin{pmatrix} 11 \\ 16 \end{pmatrix}.$$

3. Algorithms for finding robust efficient solutions in multi-objective uncertain combinatorial optimization

We now consider (MOUCO), hence, we aim to find a complete set of efficient solutions for the robust counterpart (MORCO).

3.1. Deterministic Subproblems Algorithm (DSA)

The algorithms in this section are built upon an algorithm by Bertsimas and Sim (2003) for single-objective cardinality-constrained uncertain combinatorial optimization problems, which we call Deterministic Subproblems Algorithm (DSA). Its idea is to find solutions for the uncertain problem by solving up to $|E| + 1$ deterministic problems of the same type and comparing their solutions.

In Section 3.1.1, we first describe the algorithm by Bertsimas and Sim (2003) for single-objective problems. While the authors prove correctness of the algorithm with help of duality, we provide an alternative explanation, which we later extend to (MORCO). In Section 3.1.2, we extend the algorithm for the general multi-objective case and show that the number of subproblems can be further reduced for multi-objective problems with a special property. We present several ways to reduce the number of subproblems to be solved for both the single-objective and the multi-objective case.

3.1.1. The DSA for single-objective problems

We first consider the single-objective problem $(\min_{q \in Q} z(q), c \in \mathcal{U}^{cc})$ with $\mathcal{U}^{cc}$ defined as in Eq. (1).

We now explain the algorithm by Bertsimas and Sim (2003). A solution $q \in Q$ has maximal cost (we call this its *worst case* cost), if the costs of those Γ elements, which have the largest cost intervals δ_e among all elements in q, take their maximal values $c(e) = \hat{c}_e + \delta_e$. If q has fewer than Γ elements, in the worst case the cost of all elements in q take their maximal value.

Assume that the elements are ordered with respect to the interval length δ, i.e.,

$$\bar{\delta}_1 := \delta_{e_1} \geqq \bar{\delta}_2 := \delta_{e_2} \geqq \cdots \geqq \bar{\delta}_{|E|} := \delta_{e_{|E|}} \geqq \bar{\delta}_{|E|+1} := 0.$$

For each $l \in \{1, \ldots, |E| + 1\}$ we define the function g^l (see Bertsimas & Sim, 2003):

$$g^l(q) := \sum_{e \in q} \hat{c}_e + \Gamma \cdot \bar{\delta}_l + \sum_{\substack{e_j \in q \\ j \leqq l}} (\delta_{e_j} - \bar{\delta}_l).$$

The function $g^l(q)$ is an approximation of the worst case costs of the set q. It contains

- the nominal cost $\hat{c}_e$ for each element $e \in q$, which has to be paid also in the worst case,
- $\bar{\delta}_l \cdot \Gamma$ since, in the worst case, the interval length δ_e has to be added to the costs for (at most) Γ elements,
- the positive summand $\max\{0, \delta_e - \bar{\delta}_l\}$ for each element $e \in q$ to account for all elements in the set with higher interval lengths than $\bar{\delta}_l$.

The idea of the algorithm by Bertsimas and Sim (2003) is to solve all problems

$$(\mathcal{P}(l)) \quad \min_{q \in Q} g^l(q)$$

https://doi.org/10.1016/j.ejor.2017.12.018

for $l = 1, \dots |E| + 1$ and compare the worst case values of all obtained solutions to choose a solution with minimal worst case cost. Instead of computing the worst case cost vectors, it is even sufficient to compare the objective values $g^l(q)$ of the obtained solutions and choose the solution with minimal objective value. This idea works due to the following two properties:

1. For every set q and every $l \in \{1, \dots, |E| + 1\}$ we have that $g^l(q)$ is always greater than or equal to the worst case cost $z^{\mathrm{R}}(q)$.
2. For every set q there exists some $l \in \{1, \dots, |E| + 1\}$ such that $g^l(q)$ equals the worst case cost $z^{\mathrm{R}}(q)$.

To show the first property, let q be a set and let $\{e_{a_1}, \dots, e_{a_h}\}$ be a subset of h elements in q with the largest cost intervals, where $h = \min\{|q|, \Gamma\}$. Then $z^{\mathrm{R}}(q) = \sum_{e \in q} \hat{c}_e + \sum_{j=1}^{h} \delta_{e_{a_j}}$ and we get

$$g^l(q) \geqq \sum_{e \in q} \hat{c}_e + \sum_{j=1}^{h} \bar{\delta}_l + \sum_{j=1}^{h} \max\{0, \delta_{e_{a_j}} - \bar{\delta}_l\} \geqq z^{\mathrm{R}}(q).$$

For the second property we show that for each set q there exists at least one index l with $g^l(q) = z^{\mathrm{R}}(q)$: If q has less than Γ elements, then

$$g^{|E|+1}(q) = \sum_{e \in q} \hat{c}_e + \Gamma \cdot 0 + \sum_{e \in q} (\delta_e - 0) = z^{\mathrm{R}}(q).$$

If q has at least Γ elements, let $e_{\bar{l}}$ be the element in q with the Γth smallest index. Then the Γ elements $\{e_j \in q : j \leqq \bar{l}\}$ have the largest cost intervals in q and it follows that

$$\begin{aligned} g^{\bar{l}}(q) &= \sum_{e \in q} \hat{c}_e + \Gamma \cdot \bar{\delta}_{\bar{l}} + \sum_{\substack{e_j \in q \\ j \leqq \bar{l}}} (\delta_{e_j} - \bar{\delta}_{\bar{l}}) \\ &= \sum_{e \in q} \hat{c}_e + \sum_{\substack{e_j \in q \\ j \leqq \bar{l}}} \bar{\delta}_{\bar{l}} + \sum_{\substack{e_j \in q \\ j \leqq \bar{l}}} (\delta_{e_j} - \bar{\delta}_{\bar{l}}) = z^{\mathrm{R}}(q). \end{aligned}$$

Having these two properties, we see that a robust optimal solution q^* is optimal for the problem $(\mathcal{P}(\bar{l}))$, since none of the other sets $q \in Q$ can have a better objective value. Therefore, at least one robust optimal solution will be found by the algorithm.

Algorithm 1 shows the basic structure of the described algorithm. First, the elements are ordered with respect to their interval lengths. Then the subproblems defined above are solved. Finally, of all obtained solutions the one with minimal objective value w.r.t. the respective subproblem is chosen.

Algorithm 1 Basic structure of DSA (based on Bertsimas & Sim, 2003).

Input: an instance $I = (E, Q, \hat{\boldsymbol{c}}, \boldsymbol{\delta}, \Gamma)$ of (MOUCO) with $k = 1$
Output: a robust efficient solution for I
1: Sort E w.r.t. δ_e such that $\bar{\delta}_1 := \delta_{e_1} \geqq \bar{\delta}_2 := \delta_{e_2} \geqq \dots \geqq \bar{\delta}_{|E|} \geqq \bar{\delta}_{|E|+1} := 0$.
2: Set $L := \{1, \dots, |E| + 1\}$.
3: For all $l \in L$ find an optimal solution q^l for $(\mathcal{P}(l))$.
4: Compare the objective values $z^{\mathrm{R}}(q^l)$ for all $l \in L$. The solution with the smallest objective value is a robust optimal solution.

The efficiency of Algorithm 1 depends on the time complexity to solve the subproblems $(\mathcal{P}(l))$. Because the summand $\Gamma \cdot \bar{\delta}_l$ is solution-independent, a solution for $(\mathcal{P}(l))$ can be found efficiently by solving a problem of the same kind as the underlying deterministic problem with element costs

$$c^l(e_j) := \begin{cases} \hat{c}_{e_j} + (\delta_{e_j} - \bar{\delta}_l) & \text{for } j < l \\ \hat{c}_{e_j} & \text{for } j \geqq l. \end{cases} \tag{3}$$

Hence, Algorithm 1 finds a robust optimal solution in polynomial time for many combinatorial optimization problems. Examples are the minimum spanning tree and the shortest path problem.

In the following, we show how Algorithm 1 can be enhanced. It is not necessary to solve all of the $|E| + 1$ subproblems introduced above. The following three results (see Bertsimas & Sim, 2003; Lee & Kwon, 2014; Park & Lee, 2007) can be used to reduce the number of subproblems (Lemma 8): First, if two elements have the same interval length δ_e, then their associated subproblems are identical. Second, the worst case cost of a set q with at least Γ elements equals its objective value $g^l(q)$ not only for one subproblem, but for two consecutive subproblems. Therefore, we do not miss any solutions if we only solve every second problem. Third, none of the first $\Gamma - 1$ elements can be the one with the Γth smallest index for any set in Q, so their associated subproblems need not to be solved.

Lemma 8. *(Bertsimas and Sim, 2003; Lee and Kwon, 2014; Park and Lee, 2007). The number of subproblems to be solved by Algorithm 1 can be reduced to at most $\lceil \frac{|E|-\Gamma}{2} \rceil + 1$ in the following ways:*

1. *If there are several elements $e_l, \dots, e_{(l+h)}$ with the same interval length $\delta_{e_l} = \dots = \delta_{e_{l+h}}$, only one of the subproblems $\mathcal{P}(l), \dots, \mathcal{P}(l+h)$ needs to be solved (Bertsimas & Sim, 2003).*
2. *Only every second subproblem and the last subproblem need to be solved (Lee & Kwon, 2014).*
3. *It is sufficient to start with the Γth subproblem (Park & Lee, 2007).*

Depending on the solutions that are found while the algorithm is executed, we can further reduce the number of subproblems to be solved. We refer to this newly proposed enhancement as *solution checking.*

Lemma 9. *Let $1 \leqq \tilde{l} < l \leqq |E| + 1$ and let $q^{\tilde{l}}$ be an optimal solution for $\mathcal{P}(\tilde{l})$. If $q^{\tilde{l}}$ does not contain any of the elements $e_1, \dots, e_{l-1}$, then it is optimal for $\mathcal{P}(l)$.*

Proof. We can find a solution of $\mathcal{P}(l)$ by solving a problem with the deterministic costs given in (3). For these costs we have

$$\begin{aligned} &\tilde{l} \leqq l \Rightarrow \bar{\delta}_{\tilde{l}} \geqq \bar{\delta}_l \Rightarrow c^{\tilde{l}}(e_j) \leqq c^l(e_j) \ \forall\ e_j : j < \tilde{l}, \\ &j \leqq l \Rightarrow \delta_{e_j} \geqq \bar{\delta}_l \\ &\qquad \Rightarrow c^{\tilde{l}}(e_j) = \hat{c}_{e_j} \leqq \hat{c}_{e_j} + (\delta_{e_j} - \bar{\delta}_l) = c^l(e_j) \ \forall\ e_j : \tilde{l} \leqq j < l, \\ &\tilde{l} \leqq l \Rightarrow c^{\tilde{l}}(e_j) = \hat{c}_{e_j} = c^l(e_j) \ \forall\ e_j : j \geqq l. \end{aligned}$$

If $q^{\tilde{l}}$ does not contain any element e_j: $j < l$, then

$$\sum_{e \in q^{\tilde{l}}} c^l(e) = \sum_{e \in q^{\tilde{l}}} c^{\tilde{l}}(e) \leqq \sum_{e \in q} c^{\tilde{l}}(e) \leqq \sum_{e \in q} c^l(e) \ \forall\ q \in Q,$$

hence, $q^{\tilde{l}}$ is optimal for $\mathcal{P}(l)$. □

We can therefore replace Step 3 of the basic structure (Algorithm 1) with Algorithm 2.

Algorithm 2 Improved Step 3 of Algorithm 1: solve subproblems (with solution checking).

Input: $I = (E, Q, \hat{\boldsymbol{c}}, \boldsymbol{\delta}, \Gamma)$ with E ordered w.r.t. $\boldsymbol{\delta}$, $\bar{\boldsymbol{\delta}}$, an index set L of subproblems
Output: a set of solutions $\{q^l : l \in L\}$

```
l̃ := 0
for all l ∈ L in increasing order do
    if l̃ = 0 or q^l̃ contains any element in {e_1, ..., e_{l−1}} then
        Find an optimal solution q^l for (P(l)).
    else q^l := q^l̃
    end if
    l̃ := l
end for
```

https://doi.org/10.1016/j.ejor.2017.12.018

632 A. Raith et al./European Journal of Operational Research 267 (2018) 628–642

Lemma 9 does not contain any theoretical complexity result since, in the worst case, still $\lceil \frac{|E|-\Gamma}{2} \rceil + 1$ subproblems are solved. Nevertheless, the results of our experiments in Section 4 show the practical use of this improvement.

3.1.2. The DSA for multi-objective problems

In this section, we extend the DSA to multi-objective problems. The idea presented in Section 3.1.1 is still valid. A set q has maximal cost in the ith objective, if the cost of its Γ_i elements with the largest cost intervals $\delta_{e,\,i}$ take their maximal value. However, the sorting of the elements by interval lengths often results in a different order for each objective. An element that has the Γ_ith longest interval in q for all $i = 1, \ldots, k$ is not likely to exist. To ensure that the worst case vector of q equals the objective vector of a subproblem, we have to iterate through all elements for each objective independently and consider all possible combinations. The subproblems to be solved are hence constructed in the following way:

For $j = 1, \ldots, |E|, i = 1, \ldots, k$ let E^i_j be a set of the j elements with the largest intervals for the ith objective with $E^i_1 \subset E^i_2 \subset \cdots \subset E^i_{|E|} = E$. I.e., $|E^i_j| = j$ and $\delta_{e,i} \geqq \delta_{e',i} \;\; \forall \;\; e \in E^i_j, e' \in E \setminus E^i_j$. We further define $\bar{\delta}^i_j := \min_{e \in E^i_j} \delta_{e,i}$ and $\bar{\delta}^i_{|E|+1} := 0 \;\; \forall \;\; i$. For each $l = (l_1, \ldots, l_k) \in L := \{1, \ldots, |E|+1\} \times \cdots \times \{1, \ldots, |E|+1\}$ we define the problem

$(\mathcal{MP}(\boldsymbol{l}))$

$$\min_{q \in Q} \boldsymbol{g}^{\boldsymbol{l}}(q) := \begin{pmatrix} \sum_{e \in q} \hat{c}_{e,1} + \Gamma_1 \cdot \bar{\delta}^1_{l_1} + \sum_{e \in q \cap E^1_{l_1}} (\delta_{e,1} - \bar{\delta}^1_{l_1}) \\ \vdots \\ \sum_{e \in q} \hat{c}_{e,k} + \Gamma_k \cdot \bar{\delta}^k_{l_k} + \sum_{e \in q \cap E^k_{l_k}} (\delta_{e,k} - \bar{\delta}^k_{l_k}) \end{pmatrix}.$$

We are now looking for a complete set of solutions for each of the subproblems. Such a solution set can be found by solving a deterministic multi-objective problem of the same kind as the original problem. We denote the solution set that we obtain for $\mathcal{MP}(\boldsymbol{l})$ by $OPT^{\boldsymbol{l}}$.

Algorithm 3 preserves the basic structure of DSA: first, the

Algorithm 3 DSA for general multi-objective instances.

Input: an instance $I = (E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$ of (MOUCO)
Output: a complete set of robust efficient solutions for I

1: For $i := 1, \ldots, k$:sort E w.r.t. $\delta_{e,i}$ descending and save the first j elements in E^i_j for $j = 1, \ldots, |E|$. Set $E^i_{|E|+1} := E$. Set $\bar{\delta}^i_j := \min_{e \in E^i_j} \delta_{e,i} \;\forall\; j = 1, \ldots, |E|$ and $\bar{\delta}^i_{|E|+1} := 0$.
2: Determine $L = L_1 \times L_2 \times \cdots \times L_k$: $L_i := \{1, \ldots, |E|+1\} \;\; \forall \;\; i = 1, \ldots, k$.
3: For all $\boldsymbol{l} \in L$ find a complete set of efficient solutions $OPT^{\boldsymbol{l}}$ for $(\mathcal{MP}(\boldsymbol{l}))$.
4: Compare the objective vectors $\boldsymbol{z}^{\mathbf{R}}(q)$ of all solutions in $\cup_{\boldsymbol{l} \in L} OPT^{\boldsymbol{l}}$. The solutions with non-dominated objective vectors form a complete set of robust efficient solutions.

---

elements are sorted w.r.t. $\delta_{e,\,i}$ for each $i = 1, \ldots, k$. Instead of changing the indices, we store the set E^i_j of the first j elements for all $j = 1, \ldots, |E|$, because the order of the elements depends on the objective. Then the set L is determined, which contains vectors instead of scalar values. For each element in L the subproblem defined above is solved and their solutions are compared to obtain the robust efficient solutions.

Theorem 10. *Algorithm 3 finds a complete set of robust efficient solutions for (MOUCO).*

Proof. First, we show that $\boldsymbol{g}^{\boldsymbol{l}}$ never underestimates $\boldsymbol{z}^{\mathbf{R}}$ for any objective. Further, we prove that for each feasible solution q there is an $\boldsymbol{l} \in L$ with $\boldsymbol{g}^{\boldsymbol{l}}(q) = \boldsymbol{z}^{\mathbf{R}}(q)$. We conclude that Algorithm 3 finds a complete set of robust efficient solutions.

For each $q \in Q$, $\boldsymbol{l} \in L$ and $i \in \{1, \ldots, k\}$ we show $z^{\mathrm{R}}_i(q) \leqq g^{\boldsymbol{l}}_i(q)$. Let $\{e_{a_1}, \ldots, e_{a_h}\}$ be a set of h elements in q with the largest cost intervals $\delta_{e,\,i}$, where $h = \min\{|q|, \Gamma_i\}$. Then

$$\begin{aligned} z^{\mathrm{R}}_i(q) &= \sum_{e \in q} \hat{c}_{e,i} + \sum_{j=1}^{h} (\delta_{e_{a_j},i} - \bar{\delta}^i_{l_i} + \bar{\delta}^i_{l_i}) \\ &\leqq \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}^i_{l_i} + \sum_{j=1}^{h} (\delta_{e_{a_j},i} - \bar{\delta}^i_{l_i}) \text{ since } h \leqq \Gamma_i \\ &\leqq \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}^i_{l_i} + \sum_{e \in q} \max\{0, \delta_{e,i} - \bar{\delta}^i_{l_i}\} \\ &\quad \text{since } \{e_{a_1}, \ldots, e_{a_h}\} \subseteq q \\ &= \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}^i_{l_i} + \sum_{e \in q \cap E^i_{l_i}} (\delta_{e,i} - \bar{\delta}^i_{l_i}) \\ &\quad \text{since } e \in E^i_{l_i} \Rightarrow \delta_{e,i} \geqq \bar{\delta}^i_{l_i}, e \notin E^i_{l_i} \Rightarrow \delta_{e,i} \leqq \bar{\delta}^i_{l_i} \\ &= g^{\boldsymbol{l}}_i(q). \end{aligned}$$

We conclude $\boldsymbol{g}^{\boldsymbol{l}}(q) \leqq \boldsymbol{z}^{\mathbf{R}}(q)$ for all $q \in Q$ and $\boldsymbol{l} \in L$.

We show now that for every $q \in Q$ there is an $\bar{\boldsymbol{l}} \in L$ with $\boldsymbol{g}^{\bar{\boldsymbol{l}}}(q) = \boldsymbol{z}^{\mathbf{R}}(q)$. Given $q \in Q$ we construct $\bar{\boldsymbol{l}}$ as follows: For all $i \in \{1, \ldots, k\}$ with $\Gamma_i > |q|$, we set $\bar{l}_i := |E| + 1$, since

$$z^{\mathrm{R}}_i(q) = \sum_{e \in q} \hat{c}_{e,i} + \sum_{e \in q} \delta_{e,i} = \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot 0 + \sum_{e \in q} (\delta_{e,i} - 0).$$

For all $i \in \{1, \ldots, k\}$ with $\Gamma_i \leqq |q|$ we choose $\bar{l}_i$ such that $q \cap E^i_{\bar{l}_i}$ contains exactly Γ_i elements. These Γ_i elements have the largest cost intervals $\delta_{e,\,i}$ among all elements in q, i.e., the worst case cost for q is

$$\begin{aligned} z^{\mathrm{R}}_i(q) &= \sum_{e \in q} \hat{c}_{e,i} + \sum_{e \in q \cap E^i_{\bar{l}_i}} \delta_{e,i} \\ &= \sum_{e \in q} \hat{c}_{e,i} + \sum_{e \in q \cap E^i_{\bar{l}_i}} \bar{\delta}^i_{\bar{l}_i} + \sum_{e \in q \cap E^i_{\bar{l}_i}} (\delta_{e,i} - \bar{\delta}^i_{\bar{l}_i}) \\ &= \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}^i_{\bar{l}_i} + \sum_{e \in q \cap E^i_{\bar{l}_i}} (\delta_{e,i} - \bar{\delta}^i_{\bar{l}_i}) \text{ since } |q \cap E^i_{\bar{l}_i}| = \Gamma_i. \end{aligned}$$

We conclude $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{g}^{\bar{\boldsymbol{l}}}(q)$. If q is robust efficient, then there is no $q' \in Q$ with $\boldsymbol{z}^{\mathbf{R}}(q') \leq \boldsymbol{z}^{\mathbf{R}}(q)$. It follows that

$$\nexists q' \in Q : \boldsymbol{z}^{\mathbf{R}}(q') \leqq \boldsymbol{z}^{\mathbf{R}}(q) \overset{\boldsymbol{z}^{\mathbf{R}}(q') \leqq \boldsymbol{g}^{\bar{\boldsymbol{l}}}(q')}{\Rightarrow}$$
$$\nexists q' \in Q : \boldsymbol{g}^{\bar{\boldsymbol{l}}}(q') \leqq \boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{g}^{\bar{\boldsymbol{l}}}(q).$$

Therefore, q or an equivalent solution is found at least once in the algorithm. It follows that in Step 4 the objective vector of each found solution is compared to all non-dominated objective vectors, thus only robust efficient solutions remain. It follows that the output is a complete set of robust efficient solutions. □

Example 11. Consider the instance in Example 7 (Fig. 1). In Step 1 of Algorithm 3 we obtain

$$\bar{\delta}^1 = (5, 4, 3, 3, 3, 2, 1, 1, 1)^T, \bar{\delta}^2 = (5, 5, 5, 5, 5, 5, 1, 1, 1)^T$$

and for example

$$\begin{aligned} &E^1_1 = \{(v_2, v_5)\}, \;\; E^1_2 = E^1_1 \cup \{(v_2, v_4)\}, \;\; E^1_3 = E^1_2 \cup \{(v_1, v_3)\}, \\ &E^1_4 = E^1_2 \cup \{(v_3, v_4)\}, \;\; E^1_5 = E^1_4 \cup \{(v_5, v_6)\}, \;\; E^1_6 = E^1_5 \cup \{(v_3, v_5)\}, \\ &E^1_7 = E^1_6 \cup \{(v_1, v_2)\}, \;\; E^1_8 = E^1_7 \cup \{(v_2, v_3)\}, \;\; E^1_9 = E^1_8 \cup \{(v_4, v_6)\}, \\ &E^2_1 = \{(v_2, v_5)\}, \;\; E^2_2 = E^2_1 \cup \{(v_2, v_4)\}, \;\; E^2_3 = E^2_2 \cup \{(v_1, v_3)\}, \end{aligned}$$

https://doi.org/10.1016/j.ejor.2017.12.018

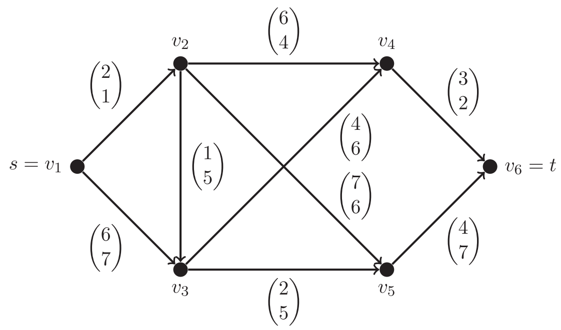


Fig. 2. $OPT^{(7,8)}$ in Example 11 is obtained by solving this instance of the multi-objective shortest path problem.

$E_4^2 = E_2^2 \cup \{(v_3, v_4)\},\ E_5^2 = E_4^2 \cup \{(v_5, v_6)\}, E_6^2 = E_5^2 \cup \{(v_3, v_5)\},$
$E_7^2 = E_6^2 \cup \{(v_1, v_2)\},\ E_8^2 = E_7^2 \cup \{(v_2, v_3)\},\ E_9^2 = E_8^2 \cup \{(v_4, v_6)\}.$

Step 3 sets $L := \{1, \dots, 9\} \times \{1, \dots, 9\}$ and in Step 3 ($\mathcal{MP}(\boldsymbol{l})$) is solved for all $\boldsymbol{l} \in L$.

As an example, we consider $\boldsymbol{l} = (7, 8)$. Recall, that the path $q_1 := ((v_1, v_2), (v_2, v_4), (v_4, v_6))$ is robust efficient. Since $|E_7^1 \cap q_1| = 2$ and $|E_8^2 \cap q_1| = 2,$ we know from the proof of Theorem 10 that $\mathbf{g^{(7,8)}}(q_1) = \boldsymbol{z}^{\mathbf{R}}(q_1)$ and that q_1 is an efficient solution for

($\mathcal{MP}(7,8)$)

$$\min_{q \in Q} \mathbf{g^{(7,8)}}(q) := \begin{pmatrix} \sum_{e \in q} \hat{c}_{e,1} + \Gamma_1 \cdot \bar{\delta}_7^1 + \sum_{e \in q \cap E_7^1} (\delta_{e,1} - \bar{\delta}_7^1) \\ \sum_{e \in q} \hat{c}_{e,2} + \Gamma_2 \cdot \bar{\delta}_8^2 + \sum_{e \in q \cap E_8^2} (\delta_{e,2} - \bar{\delta}_8^2) \end{pmatrix}.$$

A complete set of efficient solutions $OPT^{(7,8)}$ for ($\mathcal{MP}(7,8)$) can be obtained by solving the instance of the deterministic multi-objective shortest path problem shown in Fig. 2. The edge costs are

$$c_1^{(7,8)}(e) := \begin{cases} \hat{c}_{e,1} + \delta_{e,1} - \bar{\delta}_7^1 & \text{if } e \in E_7^1 \\ \hat{c}_{e,1} & \text{else} \end{cases}$$

$$c_2^{(7,8)}(e) := \begin{cases} \hat{c}_{e,2} + \delta_{e,2} - \bar{\delta}_8^2 & \text{if } e \in E_8^2 \\ \hat{c}_{e,2} & \text{else.} \end{cases}$$

The path q_1 is indeed efficient for this instance with $\mathbf{c^{(7,8)}}(q_1) = (11, 7)^T$. It follows

$$\mathbf{g^{(7,8)}}(q_1) = \begin{pmatrix} \Gamma_1 \cdot \bar{\delta}_7^1 + 11 \\ \Gamma_2 \cdot \bar{\delta}_8^2 + 7 \end{pmatrix} = \begin{pmatrix} 13 \\ 9 \end{pmatrix} = \boldsymbol{z}^{\mathbf{R}}(q_1).$$

The path $q_3 := \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_6)\}$ is efficient for this instance as well, hence $q_1, q_3 \in OPT^{(7,8)}$.

In Step 4 of Algorithm 3, all obtained solutions are compared to each other. The path q_3 is not robust efficient, because $\boldsymbol{z}^{\mathbf{R}}(q_2) = (11, 16)^T \leq (12, 16)^T = \boldsymbol{z}^{\mathbf{R}}(q_3)$. Since $q_2 \in OPT^{(4,4)}$, $\boldsymbol{z}^{\mathbf{R}}(q_2)$ and $\boldsymbol{z}^{\mathbf{R}}(q_3)$ are compared to each other in Step 4 and the returned solution set does not contain q_3. However, it contains q_1, because q_1 is robust efficient and hence there does not exist any path q' with $\boldsymbol{z}^{\mathbf{R}}(q') \leq \boldsymbol{z}^{\mathbf{R}}(q_1)$.

As for the single-objective version, we can reduce the number of subproblems to be solved. The results of Lemma 8 are still valid for each objective independently. Therefore, we can replace the L_i as described in the following lemma.

Lemma 12. *The number of subproblems to be solved by Algorithm 3 can be reduced to* $\prod_{i=1}^{k}(\lceil \frac{|E|-\Gamma_i}{2} \rceil + 1)$ *in the same ways as in the single-objective case (Lemma 8):*

1. *Let* $i \in \{1, \dots, k\}$ *be given. If there are several elements with the same interval length* $\delta_{e,i}$*, i.e., there exist pairwise different indices* $j_1, \dots, j_h \in \{1, \dots, |E|\}$ *with* $\bar{\delta}_{j_1}^i = \dots = \bar{\delta}_{j_h}^i$*, then it is sufficient that* l_i *takes one of the values in* $\{j_1, \dots, j_h\}$.
2. *For all* $i \in \{1, \dots, k\}$ *it is sufficient, that* l_i *takes every second value in* $\{1, \dots, |E|\}$ *and the value* $|E| + 1$.
3. *It is sufficient that* l_i *takes values that are greater than or equal to* Γ_i.

Proof.

1. Let $\hat{l}_1, \hat{l}_2, \dots, \hat{l}_{i-1}, \hat{l}_{i+1}, \dots, \hat{l}_k \in \{1, \dots, |E| + 1\}$ be fixed values. We define the vector $\boldsymbol{\hat{l}^x} := (\hat{l}_1, \hat{l}_2, \dots, \hat{l}_{i-1}, x, \hat{l}_{i+1}, \dots, \hat{l}_k)$. From $\bar{\delta}_{j_1}^i = \dots = \bar{\delta}_{j_h}^i$ it follows directly
$$\sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{j_1}^i + \sum_{e \in q \cap E_{j_1}^i} (\delta_{e,i} - \bar{\delta}_{j_1}^i)$$
$$= \dots = \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{j_h}^i + \sum_{e \in q \cap E_{j_h}^i} (\delta_{e,i} - \bar{\delta}_{j_h}^i)$$
and therefore $\mathcal{MP}(\boldsymbol{\hat{l}^{j_1}}) = \dots = \mathcal{MP}(\boldsymbol{\hat{l}^{j_h}})$.
2. Let $q \in Q$ be a feasible solution. We have shown in the proof of Theorem 10 that there exists an $\boldsymbol{\bar{l}} \in L$ with $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{g}^{\boldsymbol{\bar{l}}}(q)$ and either $\bar{l}_i = |E| + 1$ or $\Gamma_i = |q \cap E_{\bar{l}_i}^i|$. In the second case, since $q \cap E_{\bar{l}_i}^i$ contains the Γ_i elements in q with the largest cost intervals $\delta_{e,i}$, we have
$$z_i^{\mathrm{R}}(q) = \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{\bar{l}_i}^i + \sum_{e \in q \cap E_{\bar{l}_i}^i} (\delta_{e,i} - \bar{\delta}_{\bar{l}_i}^i)$$
$$= \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{\bar{l}_i}^i + \sum_{e \in q \cap E_{\bar{l}_i}^i} (\delta_{e,i} - \bar{\delta}_{\bar{l}_i}^i) + \Gamma_i \cdot (\bar{\delta}_{(\bar{l}_i+1)}^i - \bar{\delta}_{\bar{l}_i}^i)$$
$$+ \Gamma_i \cdot (\bar{\delta}_{\bar{l}_i}^i - \bar{\delta}_{(\bar{l}_i+1)}^i)$$
$$= \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{(\bar{l}_i+1)}^i + \sum_{e \in q \cap E_{\bar{l}_i}^i} (\delta_{e,i} - \bar{\delta}_{(\bar{l}_i+1)}^i),$$
because $|q \cap E_{\bar{l}_i}^i| = \Gamma_i$
$$= \sum_{e \in q} \hat{c}_{e,i} + \Gamma_i \cdot \bar{\delta}_{(\bar{l}_i+1)}^i + \sum_{e \in q \cap E_{(\bar{l}_i+1)}^i} (\delta_{e,i} - \bar{\delta}_{(\bar{l}_i+1)}^i),$$
because $\delta_{e,i} = \bar{\delta}_{(\bar{l}_i+1)}^i$ for $e \in E_{(\bar{l}_i+1)}^i \setminus E_{\bar{l}_i}^i$. Therefore, if $\Gamma_i \leqq |q|$, it is sufficient that l_i either takes the value $\bar{l}_i$ or $\bar{l}_i + 1$. If $\Gamma_i \geqq |q|$, it is sufficient that l_i takes the value $|E| + 1$.
3. In the proof of Theorem 10 we have show that for every $q \in Q$ there is an $\boldsymbol{\bar{l}} \in L$ with $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{g}^{\boldsymbol{\bar{l}}}(q)$ and either $\bar{l}_i = |E| + 1$ or $\Gamma_i = |q \cap E_{\bar{l}_i}^i| \leqq |E_{\bar{l}_i}^i| = \bar{l}_i$.

From statement 3 we know that l_i takes at most $|E| + 1 - (\Gamma_i - 1)$ different values. From statement 2 it follows that of these values the last one and every second of the other ones are sufficient. This leads to at most

$$\left\lfloor \frac{|E| + 1 - (\Gamma_i - 1) - 1}{2} \right\rfloor + 1 = \left\lfloor \frac{|E| - \Gamma_i + 1}{2} \right\rfloor + 1$$
$$= \left\lceil \frac{|E| - \Gamma_i}{2} \right\rceil + 1$$

different values of l_i. Therefore, it is sufficient to solve $\prod_{i=1}^{k}(\lceil \frac{|E|-\Gamma_i}{2} \rceil + 1)$ subproblems. □

Here again, we can use solution checking, i.e., skip some additional subproblems, depending on the solutions found so far. However, we now have to ensure that $\boldsymbol{\bar{l}} \leq \boldsymbol{l}$ and that none of the

https://doi.org/10.1016/j.ejor.2017.12.018

solutions in $OPT^{\tilde{\mathbf{l}}}$ contains any of the elements, whose costs have been increased.

Lemma 13. *Let $\mathbf{l}, \tilde{\mathbf{l}} \in \mathbb{Z}^k$ be given with $\tilde{\mathbf{l}} \leq \mathbf{l}$ and let J be the set of indices i with $\tilde{l}_i < l_i$. Let $OPT^{\tilde{\mathbf{l}}}$ be a complete set of efficient solutions for $\mathcal{MP}(\tilde{\mathbf{l}})$. If none of the sets in $OPT^{\tilde{\mathbf{l}}}$ contains an element in $\cup_{i \in J} E^i_{l_i}$, then $OPT^{\tilde{\mathbf{l}}}$ is a complete set of efficient solutions for $\mathcal{MP}(\mathbf{l})$.*

Proof. Since $\Gamma_i \cdot \bar{\delta}^{l_i}_i$ are solution independent constants, the minimization problem to be solved is a deterministic multi-objective problem with costs $\mathbf{c}^{\mathbf{l}}(e) = (c^{\mathbf{l}}_1(e), \ldots, c^{\mathbf{l}}_k(e))$:

$$c^{\mathbf{l}}_i(e) := \begin{cases} \hat{c}_{e,i} + (\delta_{e,i} - \bar{\delta}^i_{l_i}) & \text{for } e \in E^i_{l_i} \\ \hat{c}_{e,i} & \text{else.} \end{cases}$$

Since $\tilde{l}_i \leqq l_i \Rightarrow \bar{\delta}^i_{\tilde{l}} \geqq \bar{\delta}^i_{l}$, it follows

$$c^{\tilde{\mathbf{l}}}_i(e) = c^{\mathbf{l}}_i(e) \;\forall\; i \text{ with } l_i = \tilde{l}_i, \;\forall\; e \in E$$

$$c^{\tilde{\mathbf{l}}}_i(e) = c^{\mathbf{l}}_i(e) \;\forall\; i \text{ with } \tilde{l}_i < l_i, \;\forall\; e \in E \setminus E^i_{l_i}$$

$$c^{\tilde{\mathbf{l}}}_i(e) \leqq c^{\mathbf{l}}_i(e) \;\forall\; i, \;\forall\; e \in E.$$

Hence, if none of the sets in $OPT^{\tilde{\mathbf{l}}}$ contains any element in $\cup_{i \in J} E^i_{l_i}$, we have $c^{\tilde{\mathbf{l}}}_i(e) = c^{\mathbf{l}}_i(e)$ for all elements that are contained in any set in $OPT^{\tilde{\mathbf{l}}}$, and $c^{\tilde{\mathbf{l}}}_i(e) \leqq c^{\mathbf{l}}_i(e)$ for all elements in E. It follows, that every $q \in OPT^{\tilde{\mathbf{l}}}$ is also efficient w.r.t $\mathbf{c}^{\mathbf{l}}$. Furthermore, for every $q' \notin OPT^{\tilde{\mathbf{l}}}$ exists a $q \in OPT^{\tilde{\mathbf{l}}}$ with

$$\sum_{e \in q} \mathbf{c}^{\mathbf{l}}(e) = \sum_{e \in q} \mathbf{c}^{\tilde{\mathbf{l}}}(e) \leqq \sum_{e \in q'} \mathbf{c}^{\tilde{\mathbf{l}}}(e) \leqq \sum_{e \in q'} \mathbf{c}^{\mathbf{l}}(e),$$

so q' is either dominated w.r.t. $\mathbf{c}^{\mathbf{l}}$ or has an equivalent solution in $OPT^{\tilde{\mathbf{l}}}$. Therefore, $OPT^{\tilde{\mathbf{l}}}$ is a complete set of efficient solutions for $\mathcal{MP}(\mathbf{l})$. □

A fast way to use this result is to replace Step 3 of Algorithm 3 with Algorithm 4.

We loop through all $\mathbf{l} \in L$. In Lines 8 to 10, $OPT^{\mathbf{l}}$ is found for the current $\mathbf{l}$: either $(\mathcal{MP}(\mathbf{l}))$ is solved, or $OPT^{\mathbf{l}}$ is set to the solution set of an already solved subproblem. For this purpose, we store one vector $\tilde{\mathbf{l}}^h$ for each $h = 1, \ldots, k$, which is updated in Line 13 whenever the value l_h has changed, i.e. whenever l_i was increased for some $i \leqq h$ in the respective for-loop.

When l_h is increased in the for-loop, during the next execution of Line 8, we have:

$$\tilde{l}^h_i = \begin{cases} l_i & \text{for } i < h, \text{ because } \tilde{\mathbf{l}}^h \text{ was updated} \\ & \text{after the previous change of } l_i, \\ l_i - 1 & \text{for } i = h, \text{ because } l_h \text{ was increased,} \\ & \text{but } \tilde{\mathbf{l}}^h \text{ is not updated yet,} \\ 1 = l_i & \text{for } i > h, \text{ as, due to the nested for-loops, } l_i \\ & \text{is set to 1 whenever } l_h \text{ changes.} \end{cases}$$

Hence, if no set in $OPT^{\tilde{\mathbf{l}}^h}$ contains any element in $E^h_{l_h}$ the conditions of Lemma 13 are satisfied for $\tilde{\mathbf{l}} := \tilde{\mathbf{l}}^h$.

Corollary 14. *Algorithm 3 with Algorithm 4 replacing Step 3 and the construction of L (Step 2) adjusted according to Lemma 12, finds a complete set of robust efficient solutions for (MOUCO). During its execution at most $\prod_{i=1}^{k}(\lceil \frac{|E|-\Gamma_i}{2} \rceil + 1)$ deterministic subproblems have to be solved.*

For problems with the following property, the number of subproblems to be solved can be reduced significantly.

Definition 15. An instance $(E, Q, \hat{C}, \Delta, \mathbf{\Gamma})$ has *partial objective-independent element order* if there exists a subset $J := \{i_1, \ldots, i_r\} \subseteq \{1, \ldots, k\}$ with

Algorithm 4 Improved Step 3 of Algorithm 3: solve subproblems (with solution checking).

Input: an instance $I = (E, Q, \hat{C}, \Delta, \mathbf{\Gamma})$, $\bar{\delta}^j_i$ and E^i_j $\forall\; i, j \in \{1, \ldots, k\}$, an index set L of subproblems
Output: solution sets $(OPT^{\mathbf{l}}, \mathbf{l} \in L)$

1: $\tilde{\mathbf{l}}^1 := (0, \ldots, 0)$
2: $h := 1$
3: **for all** $l_1 \in L_1$ in increasing order **do**
4: **for all** $l_2 \in L_2$ in increasing order **do**
5: ...
6: **for all** $l_k \in L_k$ in increasing order **do**
7: $\mathbf{l} := (l_1, \ldots, l_k)$
8: **if** $\tilde{\mathbf{l}}^h = (0, \ldots, 0)$ or any of the sets in $OPT^{\tilde{\mathbf{l}}^h}$ contains any element in $E^h_{l_h}$ **then**
9: Find a complete set of efficient solutions $OPT^{\mathbf{l}}$ for $(\mathcal{MP}(\mathbf{l}))$.
10: **else** $OPT^{\mathbf{l}} := OPT^{\tilde{\mathbf{l}}^h}$
11: **end if**
12: **for** $i = h, \ldots, k$ **do**
13: $\tilde{\mathbf{l}}^i := \mathbf{l}$
14: **end for**
15: $h := k$
16: **end for**
17: ...
18: $h := 2$
19: **end for**
20: $h := 1$
21: **end for**

- $\Gamma_{i_1} = \Gamma_{i_2} = \cdots = \Gamma_{i_r}$ and
- there exists an order of the elements in E, such that

$\delta_{e_1,i} \geqq \cdots \geqq \delta_{e_{|E|},i} \;\forall\; i \in J.$

If $J = \{1, \ldots, k\}$, the instance has *objective-independent element order.*

Example 16. Consider an instance with $E = \{e_1, e_2, e_3\}$ and

$\boldsymbol{\delta}_{\mathbf{e_1}} = (1, 1, 1)^T, \boldsymbol{\delta}_{\mathbf{e_2}} = (3, 2, 1)^T, \boldsymbol{\delta}_{\mathbf{e_3}} = (2, 2, 1)^T.$

Then $\delta_{e_1,i} \leq \delta_{e_3,i} \leq \delta_{e_2,i} \;\;\forall\;\; i = 1, \ldots, 3,$ hence the instance has objective-independent element order. With

$\boldsymbol{\delta}_{\mathbf{e_1}} = (1, 2, 3)^T, \boldsymbol{\delta}_{\mathbf{e_2}} = (3, 2, 1)^T, \boldsymbol{\delta}_{\mathbf{e_3}} = (2, 2, 2)^T$

the instance does not have objective-independent element order, because $\delta_{e_1,1} < \delta_{e_3,1}$ and $\delta_{e_3,3} < \delta_{e_1,3}$. However, it has partial objective-independent element order, because, e.g., $\delta_{e_2,i} \leqq \delta_{e_3,i} \leqq \delta_{e_1,i}$ for $i = 2, 3$.

Lemma 17. *Let an instance $(E, Q, \hat{C}, \Delta, \mathbf{\Gamma})$ with partial objective-independent element order be given and let J be the set of indices defined in Definition 15. Then the nested for-loops changing $l_{i_1}, \ldots, l_{i_r}$ in Algorithm 4 can be replaced by a single for-loop. The number of solved deterministic subproblems in Algorithm 3 with Algorithm 4 (with replaced for-loops) as Step 3 and L adjusted according to Lemma 12 is then less than or equal to*

$$\left(\left\lceil \frac{|E| - \Gamma_{i_1}}{2} \right\rceil + 1\right) \quad \text{if } J = \{1, \ldots, k\}$$

$$\left(\left\lceil \frac{|E| - \Gamma_{i_1}}{2} \right\rceil + 1\right) \cdot \prod_{i \in \{1,\ldots,k\} \setminus \{i_1,\ldots,i_r\}} \left(\left\lceil \frac{|E| - \Gamma_i}{2} \right\rceil + 1\right) \quad \text{otherwise.}$$

Proof. In the proof of Theorem 10 we have shown that for each $q \in Q$ there exists an $\mathbf{l} \in L$ with $\mathbf{z}^{\mathbf{R}}(q) = \mathbf{g}^{\mathbf{l}}(q)$. We show that there always is such an $\mathbf{l}$ with $l_{i_1} = \cdots = l_{i_r}$.

https://doi.org/10.1016/j.ejor.2017.12.018

Since there exists an order of the elements in E such that $\delta_{e_1,i} \geqq \cdots \geqq \delta_{e_{|E|},i}\ \forall\ i \in J$, we can choose the sets E^i_j such that $E^{i_1}_j = \cdots = E^{i_r}_j\ \forall\ j = 1, \ldots, |E|$. In the proof of Theorem 10 we choose $\bar{l}_i$ such that $E^i_{\bar{l}_i} \cap q$ has exactly Γ_i elements. With $\Gamma_{i_1} = \cdots = \Gamma_{i_r}$ it follows $\bar{l}_{i_1} = \cdots = \bar{l}_{i_r}$. Hence, we have $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{g}^{\bar{l}}(q)$ and $\bar{l}_{i_1} = \cdots = \bar{l}_{i_r}$.

It follows that the nested for-loops changing $l_{i_1}, \ldots, l_{i_r}$ can be replaced by a single for-loop, which leads directly to the stated number of subproblems. □

3.2. Bottleneck approach

In the algorithms presented in the previous section, the number of subproblems that have to be solved increases with decreasing values of Γ_i. In this section we present a method whose complexity decreases with decreasing values of Γ_i. Its idea is to transfer (MOUCO) with k objectives into a deterministic combinatorial optimization problem of the same kind with $\sum_{i=1}^{k}(\Gamma_i + 1)$ objective functions, some of which are bottleneck functions instead of sum functions. The concept is particularly useful if an efficient algorithm for solving the deterministic multi-objective problem with sum and bottleneck functions is available. As an example we present such an algorithm for the shortest path problem in Section 3.2.2.

3.2.1. Bottleneck approach for cardinality-constrained uncertain combinatorial optimization problems

We first explain the approach for the single-objective uncertain problem $\left(\min_{q\in Q} z(q), c \in \mathcal{U}^{cc}\right)$ with $\mathcal{U}^{cc}$ as given in Eq. (1). The robust counterpart (MORCO) then reduces to

$$(RCO)\ \min_{q\in Q}\left(z^{\mathrm{R}}(q) = \max_{c\in\mathcal{U}^{cc}} \sum_{e\in q} c(e)\right).$$

Definition 18. For a subset $q \subseteq E$ and given interval lengths δ_e for all $e \in E$, we sort the elements in q by decreasing interval lengths and define $j\text{-}\max_{e\in q}\delta_e$ as the interval length of the jth element according to this sorting.

Theorem 19. *Every optimal solution for (RCO) is an efficient solution for the deterministic multi-objective problem*

$$(DCO)\ \min_{q\in Q}\left(\boldsymbol{z}^{\mathbf{D}}(q) := \begin{pmatrix} \sum_{e\in q}\hat{c}_e \\ \max_{e\in q}\delta_e \\ 2\text{-}\max_{e\in q}\delta_e \\ \vdots \\ \Gamma\text{-}\max_{e\in q}\delta_e \end{pmatrix}\right).$$

Proof. Recall that any feasible set $q \in Q$ has maximal cost if the cost of its Γ elements with the largest cost intervals take their maximal values. Let q be an optimal solution for (RCO). Assume that q is not efficient for (DCO). Then there exists a solution $q' \in Q$ that dominates q and it follows

$$\sum_{e\in q'}\hat{c}_e \leqq \sum_{e\in q}\hat{c}_e \text{ and}$$

$j\text{-}\max_{e\in q'}\delta_e \leqq j\text{-}\max_{e\in q}\delta_e \forall j = 1, \ldots, \Gamma$, with at least one inequality

$$\Rightarrow z^{\mathrm{R}}(q') = \sum_{e\in q'}\hat{c}_e + \sum_{j=1}^{\Gamma} j\text{-}\max_{e\in q'}\delta_e < \sum_{e\in q}\hat{c}_e + \sum_{j=1}^{\Gamma} j\text{-}\max_{e\in q}\delta_e = z^{\mathrm{R}}(q).$$

This contradicts q being optimal for (RCO). □

The reverse of Theorem 19 does not hold: there exist efficient solutions for (DCO), which are not optimal for (RCO), as the following example shows.

Example 20. Let G be a graph that consists of two disjoint paths q, q' from s to t with three edges each. Let the cost interval of all edges in q be [1, 1] and of all edges in q' be [0, 1] and let $\Gamma = 2$. Then both paths are efficient solutions for (DCO), because

$$\boldsymbol{z}^{\mathbf{D}}(q) = (3, 0, 0) \nleq (0, 1, 1) = \boldsymbol{z}^{\mathbf{D}}(q')$$
$$\text{and } \boldsymbol{z}^{\mathbf{D}}(q') = (0, 1, 1) \nleq (3, 0, 0) = \boldsymbol{z}^{\mathbf{D}}(q).$$

But only q' is robust efficient, because

$$z^{\mathrm{R}}(q') = 2 < 3 = z^{\mathrm{R}}(q).$$

Lemma 21. *A complete set of efficient solutions for (DCO) contains at least one optimal solution for (RCO).*

Proof. Let $Q' \subseteq Q$ be a complete set of efficient solutions for (DCO). Assume, that (RCO) has an optimal solution q that is not contained in Q'. According to Theorem 19, q is an efficient solution for (DCO), so Q' contains a solution q' with

$$\begin{pmatrix} \sum_{e\in q}\hat{c}_e \\ \max_{e\in q}\delta_e \\ 2\text{-}\max_{e\in q}\delta_e \\ \vdots \\ \Gamma\text{-}\max_{e\in q}\delta_e \end{pmatrix} = \begin{pmatrix} \sum_{e\in q'}\hat{c}_e \\ \max_{e\in q'}\delta_e \\ 2\text{-}\max_{e\in q'}\delta_e \\ \vdots \\ \Gamma\text{-}\max_{e\in q'}\delta_e \end{pmatrix}$$

$$\Rightarrow z^{\mathrm{R}}(q) = \sum_{e\in q}\hat{c}_e + \sum_{j=1}^{\Gamma} j\text{-}\max_{e\in q}\delta_e = z^{\mathrm{R}}(q')$$

and q' is optimal for (RCO). □

Now, we transfer this approach to the multi-objective case. For a problem with k objectives, we construct a deterministic problem with $m := \sum_{i=1}^{k}(\Gamma_i + 1)$ objectives.

Theorem 22. *Every efficient solution for the multi-objective robust counterpart (MORCO) is an efficient solution for the deterministic multi-objective problem*

$$(MODCO)\ \min_{q\in Q}\left(\boldsymbol{z}^{\mathbf{D}}(q) := \begin{pmatrix} \sum_{e\in q}\hat{c}_{e,1} \\ \max_{e\in q}\delta_{e,1} \\ 2\text{-}\max_{e\in q}\delta_{e,1} \\ \vdots \\ \Gamma_1\text{-}\max_{e\in q}\delta_{e,1} \\ \sum_{e\in q}\hat{c}_{e,2} \\ \max_{e\in q}\delta_{e,2} \\ \vdots \\ \Gamma_k\text{-}\max_{e\in q}\delta_{e,k} \end{pmatrix}\right).$$

A complete set of solutions for (MODCO) contains a complete set of solutions for (MORCO).

Proof. Let q be an efficient solution for (MORCO). Assume that q is not efficient for (MODCO). Analogously to the proof of Theorem 19, there is a solution $q' \in Q$ dominating q and it follows that $z^{\mathrm{R}}_i(q') < z^{\mathrm{R}}_i(q)$ for at least one $i \in \{1, \ldots, k\}$, which contradicts q being efficient for (MORCO).

Assume now, that $q \notin Q'$ with Q' being a complete set of efficient solutions for (MODCO). Since q is efficient for (MODCO), there is a solution $q' \in Q'$ equivalent to q w.r.t. the objective function of (MODCO) and it follows $\boldsymbol{z}^{\mathbf{R}}(q) = \boldsymbol{z}^{\mathbf{R}}(q')$ analogously to the proof of Lemma 21. □

With an algorithm to solve (MODCO) and a method to filter the obtained solutions we can now find a complete set of robust efficient solutions for the uncertain problem. In the case of a single-objective uncertain problem, Gorski, Klamroth, and Ruzika (2012) introduced an algorithm to solve (DCO).

https://doi.org/10.1016/j.ejor.2017.12.018

3.2.2. Label setting algorithm (LSA) for (MOUSP)

In this section, we show how to apply the bottleneck approach to the cardinality-constrained uncertain shortest path problem. We propose an adjustment of standard multi-objective labeling algorithms (label setting or label correcting) to find a complete set of robust efficient solutions.

Let (MOUSP) be defined as in Section 2.1, i.e., E is the edge set of a graph and Q the set of simple paths from a given start node s to a given end node t. Additionally we assume non-negative edge costs ($\boldsymbol{c}(e) \geqq 0 \ \forall \ e \in E, \boldsymbol{c} \in \mathcal{U}^{mcc}$) and adjust a label setting algorithm as an example.

We first recall the definition of a label, which is used in common multi-objective labeling algorithms. A label $l = (\boldsymbol{y}, v', l')$ at a node v consists of

- a cost vector $\boldsymbol{y}$, here $\boldsymbol{y} = (y_1, \ldots, y_m)^T$,
- a predecessor node v', and
- a predecessor label l'.

Every label at a node $v \neq s$ with predecessor node v' *represents* a path q from s to v whose last edge is (v', v). That means that its cost equals the cost of q and its predecessor label l' represents the subpath of q from s to v'. We assume here, that no parallel edges exist, such that v and v' uniquely define an edge (v', v). If parallel edges have to be considered, the respective edge can be contained in the label as well. The labels are constructed iteratively from existing labels at the predecessor nodes and can at any time be either *temporary* or *permanent*.

Algorithm 5 is a label setting algorithm for solving (MODCO)

Algorithm 5 Label setting algorithm to solve (MODCO) for the shortest path problem.

Input: an instance $I = (E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$ of (MOUSP)
Output: permanent labels at t, representing a complete set of efficient solutions for instance I of (MODCO)

1: Set $m := \sum_{i=1,\ldots,k}(\Gamma_i + 1)$.
2: Create a temporary label l_0 with cost $(0, \ldots, 0)^T$ at node s.
3: **while** there exists at least one temporary label **do**
4: Select a temporary label l' (at any node v') with minimal aggregate cost $\sum_{j=1,\ldots,m} y'_j$ and make it permanent.
5: **for all** outgoing edges (v', v) of v' **do**
6: Create a new temporary label l at v by Algorithm 6.
7: **if** the cost of l is dominated by or equal to the cost of another label at v **then**
8: Delete l.
9: **else if** l dominates any temporary labels at v **then**
10: Delete these labels.
11: **end if**
12: **end for**
13: **end while**

for the shortest path problem. It is based on the label setting algorithm by Martins (1984) for multi-objective shortest path problems, but we make the following adjustments:

1. In Step 4 a label must be chosen whose cost is not dominated by the cost of any other temporary label. In the algorithm by Martins (1984) the lexicographically smallest label is chosen. Based on Iori, Martello, and Pretolani (2010), we choose the label with the smallest aggregate cost function $\sum_{j=1,\ldots,m} y_j$ instead.
2. In multi-objective label setting algorithms with only sum functions (as considered by Martins, 1984) a new label $l = (\boldsymbol{y}, v', l')$ at v is created by adding the cost $\boldsymbol{y}'$ of the predecessor label l' to the edge cost. For min–max functions the (entry-wise) maximum of the edge cost and the predecessor label's cost is taken (see Gandibleux, Beugnies, & Randriamasy, 2006). To solve (MODCO) we need a new way to construct the labels: let $n_i := 1 + \sum_{j=1,\ldots,(i-1)}(\Gamma_j + 1)$ denote the index of the first objective of (MODCO) associated with the original objective z_i of (MORCO). For the sum objective functions, we add the nominal cost $\hat{c}_{e,i}$ of the edge $e := (v', v)$ to the corresponding predecessor cost entry y'_{n_i}. For the j-max objective functions, we compare for each objective z_i the interval length $\delta_{e,\,i}$ of e to each of the Γ_i longest interval lengths so far $y'_{n_i+1}, \ldots, y'_{n_i+\Gamma_i}$ and insert it at the right position (see Algorithm 6). We will use the following notation: $\boldsymbol{y} := \boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{\boldsymbol{e}}, \boldsymbol{\delta}_{\boldsymbol{e}})$.
3. In the algorithm by Martins (1984) a newly created label is only deleted if it is dominated by a label at the same node. We delete the new label even if another label with equal cost exists at the same node, because we are only looking for a complete set of efficient solutions. This is also the reason why we do not need to consider *hidden* labels, which Gandibleux et al. (2006) introduced for problems with bottleneck functions. Since new labels with the same cost as existing labels are immediately deleted, Algorithm 5 works even without the assumption that no cycles of cost $(0, \ldots, 0)$ exist.

Algorithm 6 Step 6 of Algorithm 5: create a new temporary label.

Input: an instance $I = (E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$, an edge $(v', v) \in E$, a label l' with cost $\boldsymbol{y}'$ at v'
Output: a new label l at v with predecessor label l'

for $i = 1, \ldots, k$ **do**
Set $n_i := 1 + \sum_{j=1,\ldots,(i-1)}(\Gamma_i + 1)$.
$y_{n_i} := y'_{n_i} + \hat{c}_{(v',v),i}$
$a := 1$
while $a \leqq \Gamma_i$ **do**
if $\delta_{(v',v),i} > y'_{n_i+a}$ **then**
$y_{n_i+a} := \delta_{(v',v),i}$
for $b := a+1, \ldots, \Gamma_i$ **do** $y_{n_i+b} := y'_{n_i+b-1}$
end for
$a := \Gamma_i + 1$
else
$y_{n_i+a} := y'_{n_i+a}$
$a := a + 1$
end if
end while
end for
Create the temporary label $l := ((y_0, \ldots, y_m)^T, v', l')$ at node v.

Example 23. We show the first steps of Algorithm 5 with the instance given in Example 7 as input.

1. In Lines 1 and 2, m is set to $(2+1) + (2+1) = 6$ and a temporary label l_0 with cost $(0, 0, 0, 0, 0, 0)^T$ is created at node v_1.
2. The label l_0 is made permanent in Line 4 and new temporary labels are created at the nodes v_2, v_3:

   l_2^1 at v_2 with cost $(2, 1, 0, 1, 1, 0)^T$ representing $\{(v_1, v_2)\}$
   l_3^1 at v_3 with cost $(4, 3, 0, 3, 5, 0)^T$ representing $\{(v_1, v_3)\}$.

   We now have one permanent label l_0 and two temporary labels l_2^1, l_3^1. The aggregated cost of l_2^1 is smaller than the aggregated cost of l_3^1.
3. Because of its smaller aggregated cost, l_2^1 is made permanent in the next iteration of Line 4. New labels are created:

   l_3^2 at v_3 with cost $(3, 1, 1, 2, 5, 1)^T$ representing $\{(v_1, v_2), (v_2, v_3)\}$
   l_4^1 at v_4 with cost $(5, 4, 1, 5, 1, 1)^T$ representing

https://doi.org/10.1016/j.ejor.2017.12.018

$\{(v_1, v_2), (v_2, v_4)\}$

l_5^1 at v_2 with cost $(5, 5, 1, 3, 5, 1)^T$ representing

$\{(v_1, v_2), (v_2, v_5)\}$.

As an example, we look at the creation of l_3^2 in detail: the cost vector of l_2^1 is $(2, 1, 0, 1, 1, 0)^T =: \boldsymbol{y}'$. We obtain

$$\boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{(v_2,v_3)}, \boldsymbol{\delta}_{(v_2,v_3)}) = \begin{pmatrix}2\\1\\0\\1\\1\\0\end{pmatrix} \oplus \left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1\\5\end{pmatrix}\right)$$

$$= \begin{pmatrix} y'_1 + \hat{c}_{(v_2,v_3),1} \\ y'_2 \\ \delta_{(v_2,v_3),1} \\ y'_4 + \hat{c}_{(v_2,v_3),2} \\ \delta_{(v_2,v_3),2} \\ y'_5 \end{pmatrix} = \begin{pmatrix}3\\1\\1\\2\\5\\1\end{pmatrix},$$

because $y'_2 \geqq \delta_{(v_2,v_3),1} > y'_3$ and $\delta_{(v_2,v_3),2} > y'_5$. The cost vectors of the two labels l_3^1, l_3^2 at v_3 are compared to each other. As none dominates the other, both are kept. The labels l_0, l_2^1 are now permanent. We have four temporary labels $l_3^1, l_3^2, l_4^1, l_5^1$, among which l_3^2 has the smallest aggregated cost.

After several iterations of Lines 4–13, there do not exist any temporary labels. Algorithm 5 returns 3 permanent labels at node v_6:

one with cost $(8, 4, 1, 7, 1, 1)^T$ representing

$q_1 = \{(v_1, v_2), (v_2, v_4), (v_4, v_6)\}$,

one with cost $(6, 3, 2, 6, 5, 5)^T$ representing

$q_2 = \{(v_1, v_2), (v_2, v_3), (v_3, v_5), (v_5, v_6)\}$,

one with cost $(8, 3, 1, 6, 5, 5)^T$ representing

$q_3 = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_6)\}$.

In Algorithm 7 non-dominated paths according to their worst case cost will be identified from the obtained labels, see Example 29.

Algorithm 7 LSA for the shortest path problem with cardinality-constrained uncertainty.

Input: an instance $I = (E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$ of (MOUSP)
Output: a complete set of robust efficient solutions for I

Solve (MODCO) with Algorithm 5.

For every permanent label l in t compute the worst case costs $\boldsymbol{z}^{\mathbf{R}}(q)$ of its represented path q by $z_i^{\mathrm{R}}(q) := \sum_{i=n_i,\dots,n_i+\Gamma_i} y_i$ and choose the non-dominated ones.

Obtain the represented paths by backtracking the predecessor labels.

Lemma 24. *In Algorithm 5 for every label $l = (\boldsymbol{y}, v', l')$ at a node v there exists a path q from s to v with $\boldsymbol{y} = \boldsymbol{z}^{\mathbf{D}}(q)$.*

Proof. We show the statement by induction:

The first label has cost $(0, \dots, 0)$ and represents the path only consisting of node s.

Let $\boldsymbol{y}' = (y'_1, \dots, y'_m)$ be the cost of the predecessor label l' and assume that $\boldsymbol{y}'$ equals the cost $\boldsymbol{z}^{\mathbf{D}}(q')$ of a path q' from s to v'. Let $q := q' \cup (v', v)$. Then we have

$$\forall i = 1, \dots, k: y_{n_i} = y'_{n_i} + \hat{c}_{(v',v),i} = \sum_{e \in q'} \hat{c}_{e,i} + \hat{c}_{(v',v),i} = \sum_{e \in q} \hat{c}_{e,i}.$$

Further, we distinguish two cases for all $i = 1, \dots, k$:

- Case 1: $\delta_{(v',v),i} \leqq y'_{n_i+a} \forall\, a = 1, \dots, \Gamma_i$. In this case the Γ_i edges e with biggest intervals $\delta_{e,i}$ of q' and $q' \cup (v', v)$ are the same and $y_{n_i+a} = y'_{n_i+a}$ for all $a = 1, \dots, \Gamma_i$. Therefore, $(y_{n_i}, \dots, y_{n_i+\Gamma_i}) = (z^{\mathrm{D}}_{n_i}(q), \dots, z^{\mathrm{D}}_{n_i+\Gamma_i}(q))$.
- Case 2: Either $\delta_{(v',v),i} > y'_{n_i+a}$ for $a = 1$ or $\exists\, a \in \{2, \dots, \Gamma_i\}$ with $y'_{n_i+a-1} \geqq \delta_{(v',v),i} > y'_{n_i+a}$. Then

$$\forall\ b < a: \ y_{n_i+b} = y'_{n_i+b} \text{ and } b\text{-}\max_{e \in q'} \delta_{e,i} = b\text{-}\max_{e \in q' \cup (v',v)} \delta_{e,i}$$

$$\text{for } \ b = a: \ y_{n_i+b} = \delta_{(v',v),i} = b\text{-}\max_{e \in q' \cup (v',v)} \delta_{e,i}$$

$$\forall\ b: \ \text{with } \Gamma_i \geqq b > a: \ y_{n_i+b} = y'_{n_i+b-1} = b\text{-}\max_{e \in q \cup (v',v)} \delta_{e,i}$$

It follows $(y_1, \dots, y_m) = \boldsymbol{z}^{\mathbf{D}}(q)$. □

In the deterministic case with only sum functions, subpaths of efficient paths are efficient as well, which plays an important role in the proof of Martin's algorithm. If some of the objective functions are bottleneck functions, this property does not hold any more (Gandibleux et al., 2006). In our case, since we only look for a complete set of efficient solutions, the weaker property given in Lemma 26 is sufficient (this was observed but not proven by Iori et al. (2010)).

We use the following notation to specify subpaths.

Notation 25. Let q be a simple path and v, w two nodes on q (v before w). Let then $q_{v,w}$ denote the part of q from node v to node w.

Lemma 26. *Let q from s to t be an efficient path with respect to $\boldsymbol{z}^{\mathbf{D}}$ and v, w two nodes on q (v before w). Then either $q_{v,w}$ is an efficient path from v to w or there exists an efficient path p from v to w such that $q' := q_{s,v} \cup p \cup q_{w,t}$ is equivalent to q.*

Proof. Assume that $q_{v,w}$ is not efficient w.r.t $\boldsymbol{z}^{\mathbf{D}}$. Then there exists an efficient path p from v to w that dominates $q_{v,w}$. We have

$$\sum_{e \in q'} \hat{\boldsymbol{c}}_{\boldsymbol{e}} = \sum_{e \in q_{s,v}} \hat{\boldsymbol{c}}_{\boldsymbol{e}} + \sum_{e \in p} \hat{\boldsymbol{c}}_{\boldsymbol{e}} + \sum_{e \in q_{w,t}} \hat{\boldsymbol{c}}_{\boldsymbol{e}} \leqq \sum_{q_{s,v}} \hat{\boldsymbol{c}}_{\boldsymbol{e}} + \sum_{e \in q_{v,w}} \hat{\boldsymbol{c}}_{\boldsymbol{e}} + \sum_{e \in q_{w,t}} \hat{\boldsymbol{c}}_{\boldsymbol{e}}$$
$$= \sum_{e \in q} \hat{\boldsymbol{c}}_{\boldsymbol{e}}.$$

As p dominates $q_{v,w}$, it follows $\forall\ i = 1, \dots, k, a = 1, \dots, \Gamma_i$: $a\text{-}\max_{e \in p} \delta_{e,i} \leqq a\text{-}\max_{e \in q_{v,w}} \delta_{e,i}$, and hence $a\text{-}\max_{e \in q'} \delta_{e,i} \leqq a\text{-}\max_{e \in q} \delta_{e,i}\ \forall\ i = 1, \dots, k, a = 1, \dots, \Gamma_i$.

It follows $\boldsymbol{z}^{\mathbf{D}}(q') \leqq \boldsymbol{z}^{\mathbf{D}}(q)$ and we conclude $\boldsymbol{z}^{\mathbf{D}}(q') = \boldsymbol{z}^{\mathbf{D}}(q)$, because q is efficient with respect to $\boldsymbol{z}^{\mathbf{D}}$. □

Theorem 27. *When Algorithm 5 (with Algorithm 6 as Step 6) stops, the permanent labels at t represent a complete set of efficient solutions for (MODCO).*

Proof. We have to show that each permanent label at t represents an efficient path from s to t and that for each efficient path q from s to t a permanent label at t representing q or an equivalent path exists.

The proof of the first part is analogous to the proof by Ehrgott (2006) of the multi-objective label setting algorithm by Martins (1984). For substituting the lexicographic order with the aggregate cost order we refer to Iori et al. (2010).

Now, we show that for each efficient path q from s to t a permanent label at t representing q or an equivalent path exists. Assume that we have an efficient path q from s to t, such that there is no permanent label l at t with label costs $\boldsymbol{y} = \boldsymbol{z}^{\mathbf{D}}(q)$. Consider the predecessor node v' of t on q. From Lemma 26 it follows that there is an efficient path p from s to v' with $\boldsymbol{z}^{\mathbf{D}}(p \cup (v', t)) = \boldsymbol{z}^{\mathbf{D}}(q)$.

If there exists a permanent label l' at v' with label costs $\boldsymbol{y}' = \boldsymbol{z}^{\mathbf{D}}(p)$, then, after it was made permanent in Line 4, a new

https://doi.org/10.1016/j.ejor.2017.12.018

label $\bar{l}$ at node t with label costs $\bar{\boldsymbol{y}} = \boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}',\boldsymbol{t})}, \boldsymbol{\delta}_{(\boldsymbol{v}',\boldsymbol{t})})$ was constructed in Line 6. It follows

$$\bar{\boldsymbol{y}} = \boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}',\boldsymbol{t})}, \boldsymbol{\delta}_{(\boldsymbol{v}',\boldsymbol{t})}) = \boldsymbol{z}^{\mathbf{D}}(p) \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}',\boldsymbol{t})}, \boldsymbol{\delta}_{(\boldsymbol{v}',\boldsymbol{t})}) = \boldsymbol{z}^{\mathbf{D}}(p \cup (v', t))$$
$$= \boldsymbol{z}^{\mathbf{D}}(q).$$

Consider the first label with cost $\boldsymbol{z}^{\mathbf{D}}(q)$ that was constructed at node t. If this label was deleted again, its cost vector is dominated, which contradicts the efficiency of q. If it was not deleted, then it was made permanent, which contradicts our assumption that no permanent label with costs $\boldsymbol{z}^{\mathbf{D}}(q)$ exists at t.

Therefore, there is no permanent label at the predecessor node v' of t with costs $\boldsymbol{y}'$ such that $\boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{\boldsymbol{e}}, \boldsymbol{\delta}_{\boldsymbol{e}}) = \boldsymbol{z}^{\mathbf{D}}(q)$. In the same way, we can show that there is no permanent label at the predecessor node v'' of v' with costs $\boldsymbol{y}''$ such that

$$\left(\boldsymbol{y}'' \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}'',\boldsymbol{v}')}, \boldsymbol{\delta}_{(\boldsymbol{v}'',\boldsymbol{v}')})\right) \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}',\boldsymbol{t})}, \boldsymbol{\delta}_{(\boldsymbol{v}',\boldsymbol{t})}) = \boldsymbol{y}' \oplus (\hat{\boldsymbol{c}}_{(\boldsymbol{v}',\boldsymbol{t})}, \boldsymbol{\delta}_{(\boldsymbol{v}',\boldsymbol{t})})$$
$$= \boldsymbol{z}^{\mathbf{D}}(q).$$

By induction it follows that there is no permanent label at node s with cost $(0, \dots, 0)$, which is a contradiction, because such a label is constructed in Line 2 of the algorithm and made permanent during the first execution of Line 4.

We conclude that for each efficient path q from s to t there exists a permanent label at t representing q or a path that is equivalent to q. Furthermore, each permanent label at t represents an efficient path from s to t. Therefore, the paths represented by the permanent labels are a complete set of efficient solutions. □

To find a a complete set of robust efficient solutions we have to filter the solutions obtained by the labeling algorithm (see Algorithm 7).

Corollary 28. *Algorithm 7 finds a complete set of robust efficient solutions for an instance $I = (E, Q, \hat{C}, \Delta, \boldsymbol{\Gamma})$ of (MOUSP) with $\hat{C}$ being entry-wise non-negative.*

Example 29. Consider the instance given in Example 7. From the permanent labels returned by Algorithm 5 (see Example 23), the worst costs of their represented paths are computed:

$$\boldsymbol{z}^{\mathbf{R}}(q_1) = (8+4+1, 7+1+1)^T = (13, 9)^T$$
$$\boldsymbol{z}^{\mathbf{R}}(q_2) = (6+3+2, 6+5+5)^T = (11, 16)^T$$
$$\boldsymbol{z}^{\mathbf{R}}(q_3) = (8+3+1, 6+5+5)^T = (12, 16)^T.$$

Since $\boldsymbol{z}^{\mathbf{R}}(q_3)$ is dominated by $\boldsymbol{z}^{\mathbf{R}}(q_2)$, only the paths q_1 and q_2 are returned by Algorithm 7.

4. Experimental evaluation

In this paper, we presented two approaches to find a complete set of robust efficient solutions for (MOUCO). DSA solves the uncertain problem, assuming that we know how to solve the deterministic multi-objective problem. To use the bottleneck approach we need a method to solve a deterministic multi-objective problem with several objective functions, some of which are sums and some of which are bottleneck functions. We introduced such an algorithm for the shortest path problem (LSA) and, hence, we test our approaches on the shortest path problem (MOUSP).

4.1. Hazardous material transportation

We test our algorithms for (MOUSP) on a hazardous material transportation instance: when transporting hazardous materials, on one hand, the shipping company wants to minimize travel time, distance or fuel costs. On the other hand, if an accident happens, environment and population are exposed to the hazardous material. Hence, another objective is to keep the risk and negative impacts of accidents to a minimum. Erkut, Tjandra, and Verter (2007) give an overview about objectives for hazardous material transportation and about approaches for estimating the risk and the impacts of an accident.

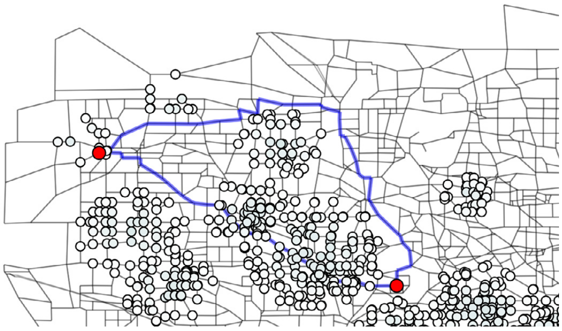


Fig. 3. Section of the Chicago regional road network with distribution of population (see Kuhn et al., 2016). The two big (red) dots show start and end node chosen for our experiments and two exemplary robust efficient paths are marked with thick (blue) lines.

For our experiments we consider the travel time and the population affected by a potential accident. We assume a nominal travel time on each road and a potential delay resulting from congestion or incidents like accidents or road construction works on some of the roads. We further assume a nominal population level, which can be increased locally by events like fairs or sport events, or due to regular shifts in population during the workday.

Our problem instance for hazardous material transportation is based on the instance used by Kuhn et al. (2016) to test an algorithm for bi-objective shortest path problems with only one uncertain objective. The underlying network (Chicago-regional) is a sector of the Chicago region road network available from Bar-Gera, Kwon, Li, and Stabler . The sector contains 1301 nodes and 4091 edges.

To obtain plausible travel times, Kuhn et al. (2016) solve a traffic assignment problem with an iterative algorithm. It models the simultaneous movement of network users, assuming travelers follow their shortest paths. Congestion effects are taken into account by a nonlinear relationship between the flow on an edge and the travel time. Until an equilibrium solution is found, each iteration of the algorithm produces a flow and resulting travel times on the edges. To obtain the lower (upper) limit of the travel time interval for each edge we choose the smallest and largest travel times obtained during several stages of the iterative equilibrium algorithm.

For the population we use the distribution of the population described by Kuhn et al. (2016) as nominal values (lower interval limits). We randomly assign integer interval lengths ($\delta_{e,2}$) up to x% of the respective nominal value. By varying x we obtain several test instances. We call x the *population uncertainty*.

We choose an appropriate start and end node with an agglomeration of population between them. Fig. 3 shows two exemplary robust efficient paths for the instance with $x = 10$ and $\boldsymbol{\Gamma} = (5, 5)$. One of the paths goes directly through the area with high population. Here the time objective function has a small value, whereas the number of people exposed to the risk of health damage in case of an accident is relatively high. The other path avoids highly populated areas, which results in a longer travel time.

https://doi.org/10.1016/j.ejor.2017.12.018

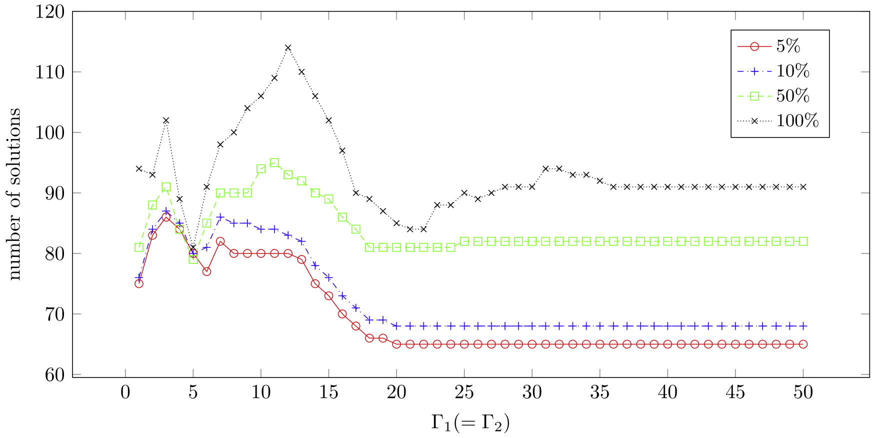


Fig. 4. Number of robust efficient solutions for several values of Γ_i and population uncertainty x.

Table 1
Number of generated solutions for several values of Γ_i and population uncertainty: Sol = minimal number of robust efficient solutions in a complete set, tDSA = total number of solutions generated in the subproblems, tLSA= total number of solutions found with the multi-objective labeling algorithm (before filtering the robust efficient solutions).

	Pop. unc. ≦5%			Pop. unc. ≦10%			Pop. unc. ≦50%			Pop. unc. ≦100%		
Γ_i	Sol	tDSA	tLSA	Sol	tDSA	tLSA	Sol	tDSA	tLSA	Sol	tDSA	tLSA
1	75	26,288	6991	76	52,887	8886	81	226,008	13189	94	468,828	16,768
2	83	26,278	4529	84	52,867	5879	88	225,928	7830	93	468,668	10,228
3	86	26,579	2972	87	53,544	3732	91	229,031	4727	102	475,140	5860
4	84	26,569	1679	85	53,524	2057	84	228,951	2184	89	474,980	2843
5	80	26,569	691	80	53,524	944	79	228,951	843	81	474,980	940
10	80	25,179	–	84	50,665	–	94	216,596	–	106	449,430	–
20	65	23,306	–	68	46,912	–	81	200,709	–	85	407,281	–
30	65	21,762	–	68	39,437	–	82	178,838	–	91	367,987	–
40	65	20,264	–	68	32,655	–	82	154,478	–	91	330,851	–
50	65	15,011	–	68	30,306	–	82	135,934	–	91	296,009	–

4.2. Results

The algorithms are implemented in C++, compiled under Debian 8.6 with g++ 4.9.2 compiler, and run on a Laptop with 2.10 gigahertz quad core processor and 7.71 gigabytes of RAM. If not stated otherwise, we use an implementation of DSA that contains all enhancements described in Section 3.1. In addition, it checks in the beginning, whether the instance has objective-independent element order. If this is the case, we use a special version of DSA, as proposed in Lemma 17, which we will refer to as DSA-oi: instead of the nested for-loops in Lines 3–6 of Algorithm 4 it only contains one for-loop.

For solving the subproblems we use an implementation of the algorithm by Martins (1984) (with the difference that the labels are selected w.r.t. their aggregate cost instead of using the lexicographic order). There and in the implementation of LSA, we additionally delete new labels at any node if they are dominated by an existing label at t.

In the figures, one data point represents one measurement, except for Section 4.2.3, where we took the average running time of 40 runs.

To compare the performance of our solution approaches, we solve the bi-objective hazardous material transportation instance described above for different values of population uncertainty x and $\mathbf{\Gamma}$. We always choose the same value for Γ_1 and Γ_2 and we will refer to this value as Γ_i in the following. In addition, we compare the performance of the algorithms on an instance with with two correlated objective functions and on an instance with three objectives. We further evaluate the improvement gained by our enhancement of DSA (solution checking). Finally, we generate an instance with objective-independent element order and investigate to which extent the performance time of the DSA benefits from the results in Lemma 17.

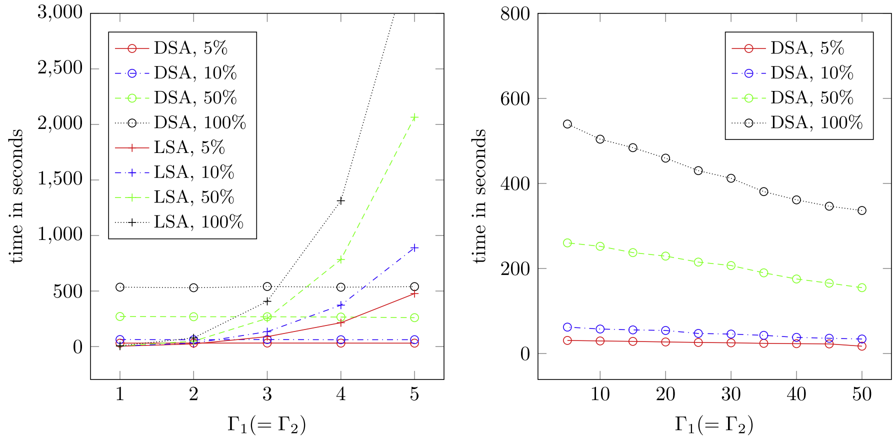
4.2.1. Number of robust efficient solutions for the hazardous material transportation instance

Fig. 4 shows the minimal cardinality of a complete set of robust efficient solutions for the generated instances for several values of x and Γ_i. In general, for increasing values of population uncertainty x the number of robust efficient solutions increases as well, because of the higher variation allowed in the second objective. We do not observe a direct dependency on Γ_i, but for values greater than 25 the number of robust efficient solutions stays the same or differs only little. The reason is that the robust efficient solutions contain only between 39 and 56 edges. Furthermore, the interval lengths $\delta_{e,1}$ resp. $\delta_{e,2}$ of some edges are 0. Hence, at some point, allowing more edges to differ from their minimal cost makes no difference.

In Table 1, we present the number of solutions generated in total: For DSA we add the number of solutions obtained by solving the subproblems (which possibly contain identical solutions several times). For LSA we list the number of solutions found by the multi-objective labeling algorithm before the filtering step. The number of solutions generated increases with the population

https://doi.org/10.1016/j.ejor.2017.12.018

640 A. Raith et al./European Journal of Operational Research 267 (2018) 628–642



(a) $\Gamma_1 = \Gamma_2 \in \{1, \ldots, 5\}$ (b) $\Gamma_1 = \Gamma_2 \in \{5, 10, \ldots, 50\}$

Fig. 5. Running time of DSA and LSA for several values of Γ_i and population uncertainty x on two different scales.

uncertainty x (as does the number of robust efficient solutions). It tends to decrease for increasing Γ_i (with a few exceptions). For the DSA that is because of the decreasing number of subproblems solved (see Fig. 8(b)).

4.2.2. Comparison of the two solution approaches

Fig. 5 shows the running time of DSA and LSA for several values of Γ_i and x. The running time of LSA increases with Γ_i, whereas the running time of DSA decreases (see also Fig. 8(a)). The reason is that for increasing Γ_i, the number of objectives in the deterministic multi-objective problem solved during LSA increases as well. However, the maximal number of subproblems solved during DSA decreases. For small values of Γ_i LSA solves the given instances faster, for higher values DSA has a better performance.

Choosing a higher value for x results in a greater maximal and mean deviation from the nominal value and a higher number of different values of $\delta_{e,2}$. When x is increased, the running time of both algorithms increases. In the case of DSA, this can be explained by the higher number of different values of $\delta_{e,2}$, which leads to a higher number of subproblems.

4.2.3. Correlated objective functions

We additionally generate an instance with two strongly correlated objective functions: we use the travel time as one objective and generate a second travel time objective by multiplying the nominal times and the interval lengths each by a random factor between 0.9 and 1.1.

Both algorithms benefit a lot from the correlation, all running times are now less than four seconds, as shown in Fig. 6. In comparison, LSA benefits more from correlated objective function values: The values of Γ_i, for which it is still faster than DSA, are much higher on this instance than on the original hazardous material transportation instance considered in Section 4.2.2. For small values of Γ_i it is much faster than DSA.

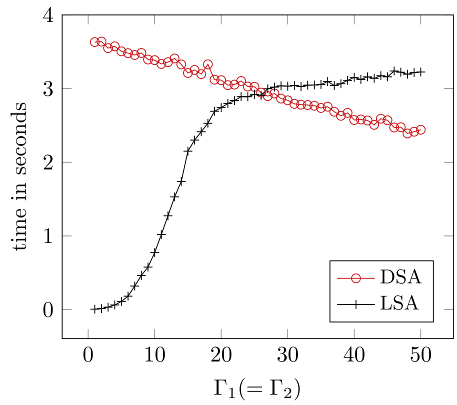


Fig. 6. Running time of DSA and LSA for an instance with two strongly correlated objective functions.

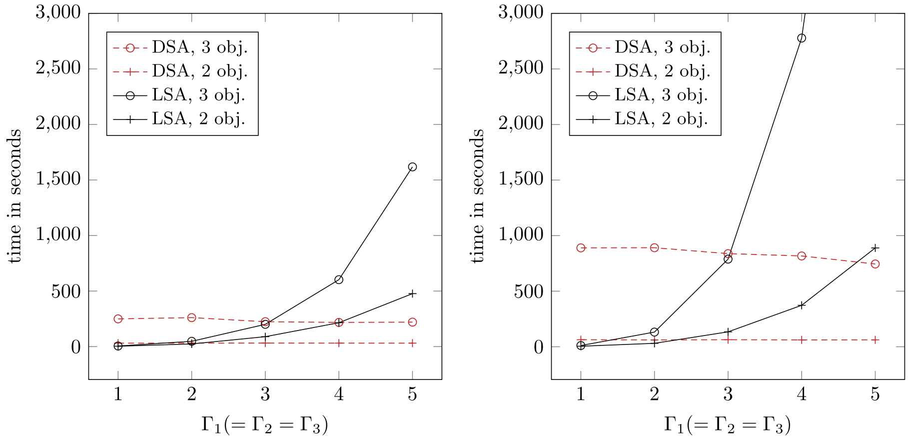
4.2.4. Three objectives

Since we are also interested in the performance of the algorithms for problems with more than two objectives, we generate an artificial third objective: For the nominal values we use again the nominal population. We generate random interval lengths in the same range as the other population objective. That means, the value of population uncertainty in general is the same for both population objectives, but the specific interval lengths of each edge may differ. Because of the identical nominal values, two of the three objectives are correlated. Fig. 7 shows the running times on this instance in comparison to the instance with two objectives described above.

The running time of both algorithms increases by including the additional objective, even though it is strongly correlated to one of the original objectives. The relative difference between the running time of the instance with two objectives and the instance with three objectives increases with Γ_i for LSA, whereas it decreases for DSA.

4.2.5. Evaluation of the improvement obtained by solution checking

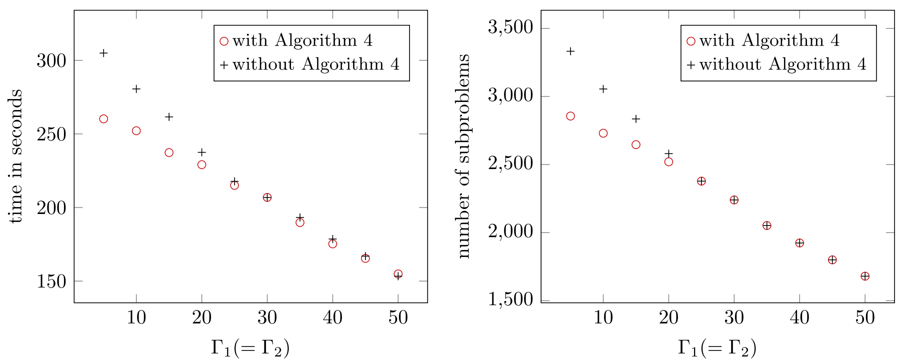
To evaluate the obtained improvement by using solution checking in DSA, we use Algorithm 4 as Step 3 of Algorithm 3. We compare the running time of the version containing solution checking to the running time of the version without this enhancement (Fig. 8(a)). Additionally, we count the solved subproblems (Fig. 8(b)). Where fewer subproblems were solved because of the enhancement, the running times differ significantly, for all other instances they are nearly equal. Hence, the check itself does



(a) Population uncertainty 5%

(b) Population uncertainty 10%

Fig. 7. Running time of DSA and LSA for an instance with three objectives and an instance with two objectives.



(a) Running time.

(b) Number of Subproblems.

Fig. 8. Running time and number of solved subproblems of DSA with and without solution checking (population uncertainty 50%).

not slow down the algorithm significantly in comparison to the acceleration that we obtain when subproblems can be skipped. We conclude that it is worth using the enhancement, but as Γ_i increases solution checking becomes less effective.

Note that, since Lemma 13 allows to exclude even more subproblems than excluded in Algorithm 4, further speed-ups may be achieved by implementing a more sophisticated solution checking. However, already when using Algorithm 4, the benefit of solution checking is clearly visible.

4.2.6. Evaluation of DSA for instances with objective-independent element order

For instances with objective-independent element order, we use the special version DSA-oi as proposed in Lemma 17. To compare its performance to the general version of DSA we construct an instance with objective-independent element order: instead of generating interval lengths for the population objective we use the interval lengths of the travel time objective. Fig. 9 shows that DSA-oi has a much better performance than the general algorithm. The test, whether the instance is objective-independent, only takes a small fraction of the running time (for our instances $1.4 \cdot 10^{-5}$ seconds). Therefore, it is reasonable to check each instance for objective-independent element order before solving it with DSA.

5. Conclusion

In this paper we developed two approaches to find minmax robust solutions for multi-objective combinatorial optimization problems with cardinality-constrained uncertainty. We extended an algorithm by Bertsimas and Sim (2003) to multi-objective optimization (DSA), suggested an enhancement and developed a special version for instances with objective-independent element order. We also introduced a second approach and used it to develop a label setting algorithm (LSA) for the multi-objective uncertain shortest path problem.

We tested our algorithms on several instances of the multi-objective uncertain shortest path problem arising from hazardous material transportation. On most of the tested instances DSA has a better performance, but LSA is faster for small values of Γ_i. If the two objective functions are strongly correlated, LSA is competitive even for higher values of Γ_i. This appears often in shortest path problems, where, e.g., the distance, travel time and fuel consumption are correlated.

When implementing DSA we recommend to use the proposed enhancements and to check whether the special version for instances with (partial) objective-independent element order can be used. The checks do not take long in comparison to the total

https://doi.org/10.1016/j.ejor.2017.12.018

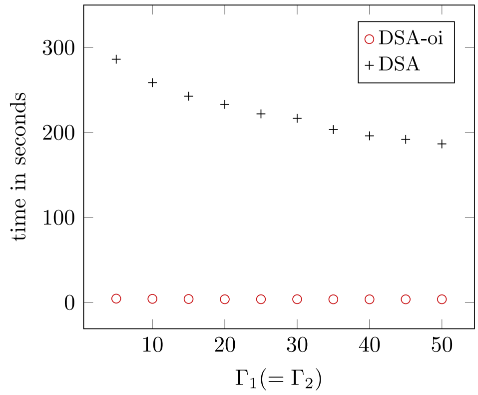


Fig. 9. Comparison of DSA and DSA-oi for instances with objective-independent element order.

running time, and if their result is positive, the algorithm can be accelerated significantly.

For further investigations other variants of multi-objective cardinality-constrained uncertainty are of interest. A second way to extend the single-objective concept is to require the edges whose costs differ from their minimal values to be the same for all objectives. In this case the uncertainties in the objectives are no longer independent of each other and using point-based or set-based minmax robust efficiency leads to different solution sets. An interesting variation of cardinality-constrained uncertainty is not to consider a bound on the cardinality, but on the sum of the deviation from their minimal values.

Further research on robust multi-objective optimization includes other types of uncertainty, e.g., discrete scenario sets or polyhedral or ellipsoidal uncertainty. Also the case of decision uncertainty, in which the solution found cannot be realized exactly, is of interest, see Eichfelder, Krüger, and Schöbel (2017) for first results.

The algorithms for the multi-objective cardinality-constrained uncertain shortest path problem presented in this paper can easily be extended to the *multi-objective single-source shortest path problem*. There, a complete set of efficient paths from a start node s to all other nodes is to be found. In the deterministic case, there exist algorithms (e.g. the algorithm by Martins, 1984) for which it can be shown that the running time is polynomial in the output size. It would be interesting to investigate whether this is the case for the uncertain problem, too.

Acknowledgment

Lisa Thom was supported by DFG RTG 1703 "Resource Efficiency in Interorganizational Networks".

References

Aissi, H., Bazgan, C., & Vanderpooten, D. (2009). Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research, 197*(2), 427–438.
Bar-Gera, H., Kwon, C., Li, J., & Stabler, B. Transportation Networks for Research Core Team. *Transportation Networks for Research*. https://github.com/bstabler/TransportationNetworks. Accessed: November, 04, 2016.
Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.
Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming, 98*(1), 49–71.
Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53.
Chuong, T. D. (2016). Optimality and duality for robust multiobjective optimization problems. *Nonlinear Analysis: Theory, Methods & Applications, 134*, 127–143.
Doolittle, E., Kerivin, H. M., & Wiecek, M. M. (2012). A robust multiobjective optimization problem with application to Internet routing. *Technical report R2012-11-DKW*. Clemson University.
Ehrgott, M. (2006). *Multicriteria optimization*. Berlin, Heidelberg: Springer.
Ehrgott, M., Ide, J., & Schöbel, A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research, 239*, 17–31.
Eichfelder, G., Krüger, C., & Schöbel, A. (2017). Decision uncertainty in multiobjective optimization. *Journal of Global Optimization, 69*, 485–510.
Erkut, E., Tjandra, S. A., & Verter, V. (2007). Hazardous materials transportation. *Handbooks in Operations Research and Management Science, 14*, 539–621.
Fliege, J., & Werner, R. (2014). Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research, 234*(2), 422–433.
Gandibleux, X., Beugnies, F., & Randriamasy, S. (2006). Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function. *4OR, 4*(1), 47–59.
Gorski, J., Klamroth, K., & Ruzika, S. (2012). Generalized multiple objective bottleneck problems. *Operations Research Letters, 40*(4), 276–281.
Hassanzadeh, F., Nemati, H., & Sun, M. (2013). Robust optimization for multiobjective programming problems with imprecise information. *Procedia Computer Science, 17*, 357–364.
Ide, J., Köbis, E., Kuroiwa, D., Schöbel, A., & Tammer, C. (2014). The relationship between multi-objective robustness concepts and set valued optimization. *Fixed Point Theory and Applications, 2014*(83), 1–20. doi:10.1186/1687-1812-2014-83.
Ide, J., & Schöbel, A. (2016). Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts. *OR Spectrum, 38*(1), 235–271.
Ide, J., Tiedemann, M., Westphal, S., & Haiduk, F. (2015). An application of deterministic and robust optimization in the wood cutting industry. *4OR, 13*(1), 35–57.
Iori, M., Martello, S., & Pretolani, D. (2010). An aggregate label setting policy for the multi-objective shortest path problem. *European Journal of Operational Research, 207*(3), 1489–1496.
Kalantari, M., Dong, C., & Davies, I. J. (2016). Multi-objective robust optimisation of unidirectional carbon/glass fibre reinforced hybrid composites under flexural loading. *Composite Structures, 138*, 264–275.
Kuhn, K., Raith, A., Schmidt, M., & Schöbel, A. (2016). Bicriteria robust optimization. *European Journal of Operational Research, 252*, 418–431.
Kuroiwa, D., & Lee, G. M. (2012). On robust multiobjective optimization. *Vietnam Journal of Mathematics, 40*(2–3), 305–317.
Lee, T., & Kwon, C. (2014). A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR, 12*(4), 373–378.
Martins, E. Q. V. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research, 16*(2), 236–245.
Park, K.-C., & Lee, K.-S. (2007). A note on robust combinatorial optimization problem. *Management Science and Financial Engineering, 13*(1), 115–119.
Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research, 21*(5), 1154–1157.
Wiecek, M. M., & Dranichak, G. M. (2016). Robust multiobjective optimization for decision making under uncertainty and conflict. In *Optimization challenges in complex, networked and risky systems* (pp. 84–114). INFORMS.
Yu, H., & Liu, H. M. (2013). Robust multiple objective game theory. *Journal of Optimization Theory and Applications, 159*(1), 272–280.

https://doi.org/10.1016/j.ejor.2017.12.018

A.2. Extensions of Labeling Algorithms for Multi-Objective Uncertain Shortest Path Problems

published in *Networks* (Raith et al., 2018a)

Authors: Andrea Raith, Marie Schmidt, Anita Schöbel, Lisa Thom

Received: 14 February 2018 | Accepted: 14 February 2018

DOI: 10.1002/net.21815

SPECIAL ISSUE ARTICLE

WILEY

Extensions of labeling algorithms for multi-objective uncertain shortest path problems

Andrea Raith[1] | **Marie Schmidt**[2] | **Anita Schöbel**[3] | **Lisa Thom**[3]

[1]Department of Engineering Science, The University of Auckland, Auckland 1142, New Zealand

[2]Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University Rotterdam, 3000 DR Rotterdam, The Netherlands

[3]Institut für Numerische und Angewandte Mathematik, Georg-August-Universität Göttingen, Göttingen 37083, Germany

Correspondence
Marie Schmidt, Department of Technology and Operations Management, RSM, Erasmus University Rotterdam, PO Box 1738, 3000 DR Rotterdam, The Netherlands.
Email: schmidt2@rsm.nl

Funding information
Deutsche Forschungsgesellschaft (DFG), Grant/Award Number: 172586392

Abstract

We consider multi-objective shortest path problems in which the edge lengths are uncertain. Different concepts for finding so-called robust efficient solutions for multi-objective robust optimization exist. In this article, we consider multi-scenario efficiency, flimsily and highly robust efficiency, and point-based and set-based min-max robust efficiency. Labeling algorithms are an important class of algorithms for multi-objective (deterministic) shortest path problems. We analyze why it is, for most of the considered concepts, not straightforward to use labeling algorithms to find robust efficient solutions. We then show two approaches to extend a generic multi-objective label correcting algorithm for these cases. We finally present extensive numerical results on the performance of the proposed algorithms.

KEYWORDS

finite uncertainty, label correcting algorithm, multi-objective optimization, multi-objective robust optimization, robust optimization, shortest path problem

1 | INTRODUCTION

We consider the well-known shortest path problem in terms of the recent field of multi-objective robust optimization, which combines concepts of multi-objective optimization and robust optimization.

In *multi-objective optimization*, several (conflicting) objectives are optimized simultaneously. For example, when transporting hazardous material, one wants to minimize the travel time, the expenses and the risk for the environment and the inhabitants of the region at the same time. In multi-objective optimization one usually tries to find *(Pareto) efficient* solutions, which cannot be improved in one objective without worsening them in another objective.

Robust optimization is one approach to deal with uncertain parameters. In particular in practical applications, usually not all parameters of an optimization problem are reliably predictable. The travel time in a road network, for example, depends on the traffic congestion and the weather. In robust optimization one wants to hedge against (all) possible scenarios, for example, by considering the worst case for each solution.

During the last decade, concepts of multi-objective and robust optimization have been combined to multi-objective robust optimization, where multiple objectives with uncertain parameters are considered. Several concepts to define *robust efficient*

https://doi.org/10.1002/net.21815

solutions have been developed; for a recent overview see [24, 41]. In this article, we consider five different concepts of robust efficiency: A solution is *multi-scenario efficient* [7, 18] if it cannot be improved for one scenario without worsening it in another scenario. *Flimsily* (resp. *highly*) *robust efficient* solutions [24] are efficient for at least one scenario (resp. for all scenarios). *Point-based* [27] and *set-based* [17] *robust efficiency* generalize the single-objective concept of minmax robustness, where the worst case for each solution is considered. The concepts can also be generalized to other cones than the standard Pareto cone [23].

The shortest path problem has been extensively investigated, both in terms of multi-objective and robust optimization, but has so far received only little attention in the context of multi-objective robust optimization. A fast way to solve the single-objective deterministic shortest path problem is to use a labeling algorithm, for example, the label setting algorithm of Dijkstra [14] or the label correcting algorithm of Bellman and Ford [2]. Label setting algorithms can be used for nonnegative edge costs, whereas label correcting algorithms are also suitable for negative edge costs. Labeling algorithms have been generalized successfully to multi-objective optimization, for example, in [28] for nonnegative edge costs and in [12] for general edge costs. For an overview on multi-objective labeling algorithms and a computational study see [33].

In robust optimization the considered *uncertainty set* plays an important role. The edge costs can, for example, all be influenced by the same parameter, as public events or weather conditions influence the travel time on all roads in an area. They can also vary independently of each other, as traffic lights slow down the passing through each road segment individually. In this article, we consider a finite set of possible scenarios, which affect the costs of all edges. The robust shortest path problem with a finite scenario set has first been investigated in [42] for two different robustness concepts. The authors present a pseudo-polynomial algorithm, which is an extended labeling algorithm. For one of the robustness concepts, the robust shortest path problem reduces to a minmax shortest path problem, which has earlier been considered in other contexts, see [32]. Reference [15] compare the minmax robust solutions for different assumed uncertainty sets based on a discrete sample of scenarios. Other popular robustness concepts include deviation robustness (see, eg, [10, 30, 34] for results on the shortest path problem). Reference [13] uses a concept similar to Pareto efficiency to solve robust shortest path problems. For an overview on solution approaches for the robust shortest path problem with various robustness concepts and uncertainty sets, see, for example, [1, 19, 25].

The multi-objective robust shortest path problem has only been considered in few papers so far. The authors of [26] introduce a solution algorithm for combinatorial problems with two objectives, of which only one is uncertain. They assume discrete and polyhedral uncertainty sets. Combinatorial problems with so-called *cardinality-constrained uncertainty*, an uncertainty concept first introduced in [4] for single-objective problems, are considered in [38].

The remainder of this article is structured as follows: First, we give an introduction to the multi-objective shortest path problem with uncertain edge costs and present several popular concepts of robust efficiency in Section 2. In Section 3, we state conditions under which a generic multi-objective label correcting algorithm can be used to find robust efficient solutions for the multi-objective uncertain problem. In Section 4, we investigate for each of the introduced concepts of robust efficiency whether they satisfy these conditions. In case the conditions are not satisfied, we propose algorithms to find robust efficient solutions. They either extend the algorithm from Section 3 or split the problem into subproblems, which can be solved by a repeated application of this algorithm. We experimentally test and compare the developed algorithms in Section 5.

2 | MULTI-OBJECTIVE ROBUST SHORTEST PATH PROBLEMS

We first give an introduction to the multi-objective shortest path problem following [16].

Let a digraph $G = (V, E)$ with node set V and set of directed edges E, a start node $s \in V$ and a target node $t \in V$ be given. We assume that no parallel edges exist, that is, an edge e is uniquely defined by its start node v and end node v' and can be written as $e = (v, v')$. A *path* is a chain of adjacent edges in G. We say that a node v *lies on* a path if v is start or end node of one of its contained edges. A path is *simple* if it contains each node at most once. For each node $v \in V$ let $\mathcal{Q}^v$ be the set of all simple paths in G from s to v. For a simple path q and two nodes v, v' on q we denote the subpath of q from v to v' by $q_{v,v'}$.

Further, let a multi-objective cost function $c : E \to \mathbb{R}^k$ on the edges be given, that is, c assigns a cost vector to each edge $e \in E$. The cost $z(q)$ of a path q is the sum of the costs of the edges it traverses, that is, for a simple path q we have $z(q) = \sum_{e \in q} c(e)$. Two paths q, q' are called *equivalent* if they have the same start and end node and $z(q) = z(q')$. Given an instance (G, c, s, t) we define the *multi-objective shortest path problem* as

https://doi.org/10.1002/net.21815

$$(MOSP) \qquad \min_{q \in \mathcal{Q}^t} z(q) = \begin{pmatrix} z_1(q) \\ \vdots \\ z_k(q) \end{pmatrix}.$$

A solution that minimizes all objectives simultaneously does usually not exist. We hence have to explain what "min" means: We introduce the well-known concept of *efficient solutions*.

Notation **1.** For two vectors $y^1, y^2 \in \mathbb{R}^k$ we use the notation

$$y^1 \leq y^2 \Leftrightarrow y_i^1 \leqq y_i^2 \text{ for } i = 1, \dots, k \text{ and } y^1 \neq y^2,$$
$$y^1 \leqq y^2 \Leftrightarrow y_i^1 \leqq y_i^2 \text{ for } i = 1, \dots, k.$$

Furthermore, we use $\mathbb{R}^k_{\geq} = \left\{y \in \mathbb{R}^k : 0 \leq y\right\}$.

In the following, we only use the symbols $<$ (strictly less than) and $\leqq$ (less than or equal to) to compare scalars.

Definition 2. A path q' *dominates* another path q with the same start and end node if $z(q') \leq z(q)$. We also say that $z(q')$ dominates $z(q)$. A path $q \in \mathcal{Q}^t$ is an *efficient* path for (MOSP) if there is no $q' \in \mathcal{Q}^t$ such that $z(q')$ dominates $z(q)$. Then $z(q)$ is called *non-dominated*. A *complete set* of efficient paths is a set $\mathcal{Q}' \subseteq \mathcal{Q}^t$, such that for each efficient path $q \in \mathcal{Q}^t$ there exists an equivalent path $q' \in \mathcal{Q}'$.

Solving (*MOSP*) means to find a complete set of efficient paths.

Often the costs for the edges are not known exactly, but they depend on the scenario that occurs, for example, travel times can depend on the time of the day, on special events, on the weather, etc. Here, we consider multi-objective uncertain shortest path problems with a *finite* set of scenarios $\mathcal{U} := \{\xi_1, \dots, \xi_r\}$. In multi-objective uncertain optimization, the cost vectors depend on the scenario which occurs, that is, for every scenario we may get a different cost vector. Hence, c is a function that assigns a cost vector $c(e, \xi) = (c_1(e, \xi), \dots, c_k(e, \xi))^T \in \mathbb{R}^k$ to each edge $e \in E$ for each scenario $\xi \in \mathcal{U}$. We hence obtain a cost matrix

$$c(e) := \begin{pmatrix} c_1(e, \xi_1) & \dots & c_1(e, \xi_r) \\ \vdots & & \\ c_k(e, \xi_1) & \dots & c_k(e, \xi_r) \end{pmatrix} \tag{1}$$

for every edge e. The cost of a path q is the sum of the costs of the edges it traverses, that is, for a simple path we have $z(q, \xi) = \sum_{e \in q} c(e, \xi)$ and its cost matrix is $z(q) = \sum_{e \in q} c(e)$. In this setting, two paths are called *equivalent* if they have the same start and end node and $z(q) = z(q')$. For a matrix Y we denote by $Y_{(i,\cdot)}$ its i-th row and by $Y_{(\cdot,j)}$ its j-th column, that is, $c(e)_{(\cdot,j)} = c(e, \xi_j)$.

The *multi-objective uncertain shortest path problem* (*MOUSP*) is the family of multi-objective optimization problems

$$(MOUSP) \qquad \left((MOSP_\xi) \min_{q \in \mathcal{Q}^t} z(q, \xi), \xi \in \mathcal{U}\right).$$

The notion of what is a good solution to a multi-objective uncertain problem is not trivial. In multi-objective robust optimization one searches for so-called *robust efficient* solutions. We now present some concepts to define robust efficient solutions proposed in the literature.

The concept of *multi-scenario efficiency* [7, 18] applies the idea of efficiency to several scenarios and multiple objective functions at the same time: A solution is multi-scenario efficient, if there is no other solution which dominates it in one scenario and is as least as good in all other scenarios.

https://doi.org/10.1002/net.21815

Definition 3 ([7, 18])**.** A solution $q \in \mathcal{Q}^t$ is *multi-scenario efficient* for (*MOUSP*) if

$$\nexists q' \in \mathcal{Q}^t : \begin{pmatrix} z_1(q',\xi_1) \\ \vdots \\ z_k(q',\xi_1) \\ z_1(q',\xi_2) \\ \vdots \\ z_k(q',\xi_r) \end{pmatrix} \leq \begin{pmatrix} z_1(q,\xi_1) \\ \vdots \\ z_k(q,\xi_1) \\ z_1(q,\xi_2) \\ \vdots \\ z_k(q,\xi_r). \end{pmatrix}$$

Using the concept of *highly robust efficiency* [5, 24], we look for solutions, which are efficient for every scenario.

Definition 4 ([5, 24])**.** A solution $q \in \mathcal{Q}^t$ is *highly robust efficient* for (*MOUSP*) if

$$\forall \xi \in \mathcal{U} : \nexists q' \in \mathcal{Q}^t : z(q',\xi) \leq z(q,\xi).$$

However, there is no guarantee that a highly robust efficient solution exists. A reasonable condition for a good solution would then be that it should be efficient for at least one of the scenarios. This is called *flimsily robust efficiency* in [24].

Definition 5 ([5, 24])**.** A solution $q \in \mathcal{Q}^t$ is *flimsily robust efficient* for (*MOUSP*) if

$$\exists \xi \in \mathcal{U} : \nexists q' \in \mathcal{Q}^t : z(q',\xi) \leq z(q,\xi).$$

Often in robust optimization one wants to hedge against the worst case. The aim of single-objective minmax robust optimization is to find a solution with the smallest cost in the worst case. We present two generalizations of this concept to multi-objective optimization, *point-based* and *set-based minmax robust efficiency*.

Definition 6 ([27]) A solution $q \in \mathcal{Q}^t$ is *point-based minmax robust efficient* for (*MOUSP*) if it is efficient for the deterministic multi-objective problem

$$(MOSP_{\max}) \qquad \min_{q \in \mathcal{Q}^t} \begin{pmatrix} \max_{\xi \in \mathcal{U}} z_1(q,\xi) \\ \vdots \\ \max_{\xi \in \mathcal{U}} z_k(q,\xi). \end{pmatrix}$$

Definition 7 ([17])**.** A solution $q \in \mathcal{Q}^t$ is *set-based minmax robust efficient* for (*MOUSP*) if there is no feasible solution $q' \in \mathcal{Q}^t$ with

$$z_{\mathcal{U}}(q') \subseteq z_{\mathcal{U}}(q) - \mathbb{R}^k_{\geq},$$

where

$$z_{\mathcal{U}}(q) := \{z(q,\xi) : \xi \in \mathcal{U}\}.$$

We remark that more concepts for defining robust efficient solutions to a multi-objective uncertain optimization problem exist; we refer to [24, 41] for an overview.

In this article, we are interested in solving the *multi-objective robust shortest path problem* with k objective functions for a finite scenario set $\mathcal{U}$ with $|\mathcal{U}| = r$. An instance is hence given as $(G, \mathcal{U}, c, s, t)$ with G the digraph,

https://doi.org/10.1002/net.21815

s the start and t the end node, and c denoting the objective function, which assigns for each scenario a cost vector $c(e, \xi)$ to each edge.

$$(MORSP) \quad \text{Given a concept of robust efficiency, find a complete set of robust efficient solutions for } (MOUSP).$$

That is, find a complete set of multi-scenario efficient, flimsily robust efficient, highly robust efficient, point-based minmax robust efficient or set-based minmax robust efficient solutions.

For $|\mathcal{U}| = 1$, $(MOUSP)$ reduces to $(MOSP)$. In this case, the robust efficient solutions w.r.t. any of the concepts defined in this section are exactly the efficient solutions of $(MOSP)$.

3 | GENERAL LABEL CORRECTING ALGORITHM

Labeling algorithms are a standard method for solving shortest path problems, in the single-objective as well as in the multi-objective case. Label setting algorithms can be used for instances with positive edge costs, whereas label correcting algorithms also work for negative edge costs, as long as there are no negative cycles. They can be based on node selection or label selection. We consider a generic label selection method as given in [20] for the multi-objective shortest path problem (see also [9] for the bi-objective problem).

A *label* is a tuple $l = (v, z, l')$ consisting of

- a node $v \in V$ (we say that l is a label at v),
- a cost $z(l)$, and
- a predecessor label l' (or 0 if l is the start label with cost 0 at s).

Every label l at a node $v \neq s$ *represents* a path q from s to v. That means that $z(l) = z(q)$ and l's predecessor label l' represents the subpath of q from s to v', with (v', v) being the last edge of q. Given the label l, its corresponding path q can be constructed by backtracking the nodes of the predecessor labels. These labels are called *ancestors* of l.

The labels are constructed iteratively from their predecessor labels. We store them in two label sets: A newly created label is first added to the set of temporary labels $\mathcal{T}$. As soon as a label $l \in \mathcal{T}$ at a node v is chosen in the label selection step, it is stored in the label set $\mathcal{L}$ instead and, at the end nodes of all outgoing edges of v, new labels with predecessor label l are created. The cost of a label can efficiently be computed by adding the cost of the predecessor label and the edge cost. We say that a label l is *dominated by* a label l' if $z(l)$ is dominated by $z(l')$.

Algorithm 1 is a generic label correcting algorithm with label selection as given in [20], but with an adjustment: We look for a complete set and not for the whole set of efficient solutions as done in [20]. This is why we only keep newly created labels if there is not yet any other label at the same node with the same cost. That is, we only keep track of a new path if it is not equivalent to an already existing path. Label correcting algorithms are widely used for solving multi-objective shortest path problems. The goal of this article is to make use of labeling algorithms also for solving uncertain multi-objective shortest path problems, that is, to compute *robust* efficient shortest paths.

We now discuss how we can transfer Algorithm 1 to a solution algorithm for solving the multi-objective robust shortest path problem. The first difference is that in the concepts of robust efficiency given in Section 2, the set of *optimal* solutions, that is, the set of robust efficient paths in $\mathcal{Q}^t$, is defined explicitly and not implicitly via a dominance relation. However, in order to compare label costs we need a suitable definition of dominance. For the decision if a path dominates another one, all data of the uncertain problem has to be available, that is, we need cost matrices $c(e)$ given in (1) on every edge $e \in E$. Finally, Algorithm 1 can only work if Bellman's principle of optimality [2] holds for the given concept of robust efficiency.

We summarize these conditions below.

1. Principle of optimality: For every instance $(G, \mathcal{U}, c, s, t)$ of $(MOUSP)$ we require: If $q \in \mathcal{Q}^t$ is a robust efficient path for $(G, \mathcal{U}, c, s, t)$, then for every node v in q the subpath $q_{s,v}$ is robust efficient for the instance $(G, \mathcal{U}, c, s, v)$.
2. For every $k, r \in \mathbb{N}$ there exists a binary (dominance) relation $R \subseteq \mathbb{R}^{k \times r} \times \mathbb{R}^{k \times r}$ with the following properties:

https://doi.org/10.1002/net.21815

Algorithm 1 General structure of a label correcting algorithm for (MOSP)

Input: an instance $I = (G, c, s, t)$ of the multi-objective shortest path problem (*MOSP*)
Output: label set $\mathcal{L}$, of which the labels at t represent a complete set of efficient solutions of (*MOSP*)

1: Set $l_0 := (s, 0, 0)$, $\mathcal{T} := \{l_0\}$, $\mathcal{L} := \emptyset$
2: **while** $\mathcal{T} \neq \emptyset$ **do**
3: Choose a label $l' \in \mathcal{T}$ at any node v', $\mathcal{T} := \mathcal{T} \setminus \{l'\}$, $\mathcal{L} := \mathcal{L} \cup \{l'\}$
4: **for all** outgoing edges $e = (v', v)$ of v' **do**
5: Set $l := (v, z(l') + c(e), l')$.
6: **if** there is no label $\tilde{l} \in \mathcal{T} \cup \mathcal{L}$ at node v dominating l or with $z(l) = z(\tilde{l})$ **then**
7: $\mathcal{T} := \mathcal{T} \cup \{l\}$
8: **for all** labels $\tilde{l} \in \mathcal{T}$ at v dominated by l **do**
9: $\mathcal{T} := \mathcal{T} \setminus \{\tilde{l}\}$
10: **for all** labels $\tilde{l} \in \mathcal{L}$ at v dominated by l **do**
11: $\mathcal{L} := \mathcal{L} \setminus \{\tilde{l}\}$

(a) The relation is consistent with the concept of robust efficiency: For all instances with k objectives and $|\mathcal{U}| = r$:

$$q \in Q^t \text{is robust efficient } \Leftrightarrow \nexists q' \in Q^t : (z(q'), z(q)) \in R$$

(b) Domination property (see [3]): For all instances with k objectives and $|\mathcal{U}| = r$:

$$q \in Q^t \text{is not robust efficient } \Rightarrow \exists \text{ robust efficient } q' \in Q^t : (z(q'), z(q)) \in R$$

(c) R is transitive, that is, $(Y^1, Y^2) \in R, (Y^2, Y^3) \in R \Rightarrow (Y^1, Y^3) \in R$.

We say that q' *dominates* q if $(z(q'), z(q)) \in R$.

With these conditions satisfied, all structural requirements that ensured correctness of Algorithm 1 for the deterministic case are guaranteed and we easily transfer Algorithm 1 to a solution algorithm for solving the multi-objective robust shortest path problem, which we call Algorithm 1'. As input it takes an instance $(G, \mathcal{U}, c, s, t)$ of (*MORSP*) with edge costs $c(e) \in \mathbb{R}^{k \times r}$. It executes the same steps as Algorithm 1, but using the definition of dominance given in Condition 2.

To ensure that Algorithm 1' terminates we use the common requirement that the instance is *conservative w.r.t. R*, that is, for all cycles $C \in G$ either $z(C) = 0$ or $\forall\, Y \in \mathbb{R}^{k \times r} : (Y, Y + z(C)) \in R$. Note that in single-objective deterministic optimization, conservativeness requires that no cycles of negative cost exist.

Theorem 8. *If the concept of robust efficiency satisfies Conditions 1 and 2 and the instance is conservative w.r.t. R, Algorithm 1' finds a complete set of robust efficient solutions.*

Proof. We now check that Conditions 1 and 2 and the requirement of conservativeness are indeed enough to guarantee finiteness and correctness of Algorithm 1', proceeding analogously to a proof for correctness of Algorithm 1 in the deterministic case:

We first remark that a label representing a non-simple path p will never be added to $\mathcal{T}$ in Line 6: Whenever the algorithm considers adding the label corresponding to p, this label will be dominated by or have the same cost as the label l' of the corresponding simple path p'. Since p' is a subpath of p and R is transitive, either l' or a label that has the same cost as l' or dominates l' (and thus l) will already be contained in $\mathcal{T} \cup \mathcal{L}$. Since in each iteration of Line 3 at least one label is removed from $\mathcal{T}$ and there are only finitely many simple paths in G, Algorithm 1' stops after finitely many iterations.

To see that for each robust efficient path p there will be a label l with $z(l) = z(p)$ in $\mathcal{L}$ when the algorithm terminates, note that Lines 1–5 and 7 describe a routine which iteratively constructs *all* paths from the source. This routine is complemented by Lines 6, 8–11 in which dominated labels are removed. This also prevents paths with dominated subpaths to be constructed. However, Condition 2(a) guarantees that a label corresponding to a subpath

https://doi.org/10.1002/net.21815

of a robust efficient path is only removed during the dominance check if there already exists a label with the same cost. Hence for every robust efficient path p a label l with $z(l) = z(p)$ will be found. On the other hand, any label corresponding to a path which is not robust efficient will be sorted out due to Condition 2(b), so that we obtain a complete set of robust efficient paths. ■

Conditions similar to Conditions 1 and 2 are used in [35] for a labeling approach in cycle-free graphs and (partly) in earlier dynamic programming literature (eg, [8, 21, 29, 31]). The main conceptual difference is that they start with a given dominance relation and define optimality and a counterpart to Condition 1 based on this relation. We chose to state the principle of optimality in a way which does not pre-suppose the existence of a suitable dominance relation, since the concepts for robust efficiency studied in this article are not defined via a dominance relation, and it is not immediately obvious for which of the concepts a suitable dominance relation exists (see Section 4 for the corresponding analysis).

Further, instead of requiring Property 2(b), they often require asymmetry of the considered relation. Although on their own these properties are not equivalent, they are equivalent if Properties 2(a) and 2(c) hold, as we show in the following lemma.

Lemma 9. *Let R be a binary relation with Properties 2(a) and 2(c). Then Property 2(b) is equivalent to asymmetry of R, that is, to $(Y, Y') \in R \Rightarrow (Y', Y) \notin R$.*

Proof. We first show by contradiction that asymmetry of R follows from Property 2(b). Let R have Property 2(b). Assume that there exist two matrices $Y, Y' \in \mathbb{R}^{k\times r}$ with $(Y, Y') \in R$ and $(Y', Y) \in R$. We construct an instance with only two (distinct) paths q, q' from s to t with $z(q) = Y$ and $z(q') = Y'$. Then q dominates q' and vice versa. Hence, q is not robust efficient, but there exists no robust efficient path from s to t dominating q. This is a contradiction to Property 2(b). On the other hand, Property 2(b) follows from asymmetry of R due to the finiteness of the set $\mathcal{Q}^t$. This has been shown, for example, in [35, Lemma 17] for relations on the solution set, which we can define from the given relation in the objective space. ■

4 | LABELING FOR THE MULTI-OBJECTIVE ROBUST SHORTEST PATH PROBLEM

In the following we discuss whether the concepts of robust efficiency presented in Section 2 satisfy the conditions given in Section 3 for using Algorithm 1'. If a concept does not satisfy the conditions, we investigate whether and how the idea of label correcting algorithms can nevertheless be used to find robust efficient solutions.

4.1 | Multi-scenario efficiency

Recall that a solution is multi-scenario efficient if it is efficient w.r.t. the deterministic multi-objective edge costs $c(e) = (c_1(e, \xi_1), \ldots, c_k(e, \xi_1), c_1(e, \xi_2), \ldots, c_k(e, \xi_r))^T$. We can hence reduce (*MORSP*) to a deterministic multi-objective problem and directly use Algorithm 1 to solve it. Note that the set of multi-scenario efficient solutions contains all highly robust efficient solutions as well as the set of all so-called *strictly* flimsily robust efficient, *strictly* point-based and *strictly* set-based minmax robust efficient solutions [7].

4.2 | Flimsily robust efficiency

Recall that a solution is flimsily robust efficient if it is efficient for at least one scenario in $\mathcal{U}$. We show that for flimsily robust efficiency, Condition 1 for using Algorithm 1' is satisfied, but not Condition 2. We then extend Algorithm 1' by storing some additional information for each label, such that we can find a complete set of flimsily robust efficient solutions. We also introduce an alternative solution approach which finds a complete set of flimsily robust efficient solutions by applying Algorithm 1' once for each scenario and taking the union of the solution sets.

Lemma 10. *Let q be a flimsily robust efficient path for an instance $(G, \mathcal{U}, c, s, t)$ of (MOUSP). Then, for every intermediate node v on q, the subpath $q_{s,v}$ is flimsily robust efficient for $(G, \mathcal{U}, c, s, v)$, hence Condition 1 is satisfied.*

https://doi.org/10.1002/net.21815

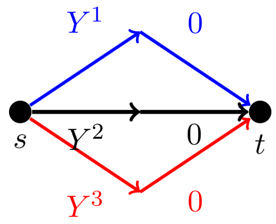


FIGURE 1 Condition 2 is not satisfied for flimsily robust efficiency (Lemma 11) [Color figure can be viewed at wileyonlinelibrary.com]

Proof. Assume that $q_{s,v}$ is not flimsily robust efficient for the instance $(G, \mathcal{U}, c, s, v)$. Then for each $\xi \in \mathcal{U}$ there exists a path $q^{\xi} \in \mathcal{Q}^{v}$ with $z(q^{\xi}, \xi) \leq z(q_{s,v}, \xi)$. From

$$z(q^{\xi}, \xi) \leq z(q_{s,v}, \xi) \Rightarrow z(q^{\xi}, \xi) + z(q_{v,t}, \xi) \leq z(q_{s,v}, \xi) + z(q_{v,t}, \xi) = z(q, \xi),$$

we conclude that for each $\xi \in \mathcal{U}$ there exists a path from s to t dominating q in scenario ξ. This contradicts q being flimsily robust efficient. ∎

The following lemma shows that for flimsily robust efficiency there does not exist a binary relation as required in Condition 2, even for only two objectives and two scenarios.

Lemma 11. *For flimsily robust efficiency and $k = r = 2$, there does not exist a binary relation with the Property 2(a) given in Condition 2.*

Proof. Assume that for $k = r = 2$ there exists a binary relation $R \subseteq \mathbb{R}^{k \times r} \times \mathbb{R}^{k \times r}$ with Property 2(a). Consider an instance of (*MOUSP*) with three disjoint paths as feasible set with the following cost matrices

$$z(\mathcal{Q}^{t}) = \left\{ Y^1 := \begin{pmatrix} 0 & 5 \\ 0 & 5 \end{pmatrix}, Y^2 := \begin{pmatrix} 1 & 4 \\ 1 & 4 \end{pmatrix}, Y^3 := \begin{pmatrix} 2 & 3 \\ 2 & 3 \end{pmatrix} \right\},$$

for example, as in Figure 1. For such an instance, a path q with $z(q) = Y^2$ is not flimsily robust efficient, because we have $Y^1_{(\cdot,1)} \leq Y^2_{(\cdot,1)}$ and $Y^3_{(\cdot,2)} \leq Y^2_{(\cdot,2)}$. It follows that $(Y^1, Y^2) \in R$ or $(Y^3, Y^2) \in R$ because of Property 2(a). However, for instances with $z(\mathcal{Q}^t) = \{Y^1, Y^2\}$ resp. $z(\mathcal{Q}^t) = \{Y^3, Y^2\}$, a path $q \in \mathcal{Q}^t$ with $z(q) = Y^2$ is flimsily robust efficient, since $Y^1_{(\cdot,2)} \nleq Y^2_{(\cdot,2)}$ and $Y^3_{(\cdot,1)} \nleq Y^2_{(\cdot,1)}$ holds. It follows that $(Y^1, Y^2) \notin R$ and $(Y^3, Y^2) \notin R$, which is a contradiction. ∎

From Lemma 11 it follows that for finding flimsily robust efficient solutions there is no suitable binary dominance relation to be used in Algorithm 1'. It is not sufficient to compare the cost matrices of the paths pairwise without considering additional information in Lines 6–11 of Algorithm 1'. However, if we store the information from previous comparisons, we can eliminate labels representing paths which are not flimsily robust efficient by pairwise comparisons.

Using this idea, we extend Algorithm 1' to Algorithm 2. For each label l we use a binary vector $x(l) \in \{0, 1\}^{|\mathcal{U}|}$ to indicate under which scenarios its path has been shown to be dominated. With q being the path represented by l we define $z(l, \xi) := z(q, \xi)$. Algorithm 2 finds a complete set of flimsily robust efficient solutions for instances where each cycle has either cost 0 for each scenario or has cost ≥ 0 for each scenario.

Note that this condition is stronger than requiring conservativeness w.r.t. $\leq$ for each scenario individually: For example, for $|\mathcal{U}| = \{\xi_1, \xi_2\}$ and a cycle C with $z(C, \xi_1) = (0, 0)^T$ and $z(C, \xi_2) = (1, 1)^T$ we have $\forall \xi \in \mathcal{U} : z(C, \xi) \geqq 0$, but neither $\forall \xi \in \mathcal{U} : z(C, \xi) \geq 0$ nor $\forall \xi \in \mathcal{U} : z(C, \xi) = 0$.

Correctness of this algorithm can be proven similarly to the proof of Theorem 8, which leads to the following theorem.

Theorem 12. *If for each cycle C in G either $\forall \xi \in \mathcal{U} : z(C, \xi) = 0$ or $\forall \xi \in \mathcal{U} : 0 \leq z(C, \xi)$, then the output label set of Algorithm 2 represents a complete set of flimsily robust efficient solutions of $(G, \mathcal{U}, c, s, t)$.*

https://doi.org/10.1002/net.21815

Algorithm 2 Extended label correcting algorithm to find flimsily robust efficient solutions

Input: an instance $I = (G, \mathcal{U}, c, s, t)$ of the multi-objective uncertain shortest path problem (*MOUSP*)
Output: label set $\mathcal{L}$, of which the labels at t represent a complete set of flimsily robust efficient solutions of (*MOUSP*)
1: Create a label l_0 with cost 0 at node s, $\mathcal{T} := \{l_0\}$, $\mathcal{L} := \emptyset$
2: **while** $\mathcal{T} \neq \emptyset$ **do**
3:   Choose a label $l' \in \mathcal{T}$ at any node v', $\mathcal{T} := \mathcal{T} \setminus \{l'\}$, $\mathcal{L} := \mathcal{L} \cup \{l'\}$
4:   **for all** outgoing edges $e = (v', v)$ of v' **do**
5:     Set $l := (v, z(l') + c(e), l')$.
6:     **for all** $\xi \in \mathcal{U}$ **do**
7:       **if** $\exists \tilde{l} \in \mathcal{T} \cup \mathcal{L}$ at v with $z(\tilde{l}, \xi) \leq z(l, \xi)$ **then**
8:         Set $x_\xi(l) := 1$.
9:     **if** $x(l) \neq (1, \ldots, 1)$ and $\nexists \tilde{l} \in \mathcal{T} \cup \mathcal{L}$ at v with $z(l) = z(\tilde{l})$ **then**
10:       $\mathcal{T} := \mathcal{T} \cup \{l\}$
11:       **for all** $\xi \in \mathcal{U}$ **do**
12:         **for all** $\tilde{l} \in \mathcal{T} \cup \mathcal{L}$ at v with $z(l, \xi) \leq z(\tilde{l}, \xi)$ **do**
13:           Set $x_\xi(\tilde{l}) := 1$.
14:           **if** $x(\tilde{l}) = (1, \ldots, 1)$ and $\tilde{l} \in \mathcal{T}$ **then**
15:             $\mathcal{T} := \mathcal{T} \setminus \{\tilde{l}\}$
16:           **if** $x(\tilde{l}) = (1, \ldots, 1)$ and $\tilde{l} \in \mathcal{L}$ **then**
17:             $\mathcal{L} := \mathcal{L} \setminus \{\tilde{l}\}$

Proof. Note that a label l is deleted if and only if its corresponding vector $x(l)$ contains only ones, that is, if l is dominated in each scenario.

Since each cycle has either cost 0 or has costs ≥ 0 for each scenario, the cost of a non-simple path is either equal to the cost of the corresponding simple path or dominated by it in each scenario. Analogous to the proof of Theorem 8, whenever the algorithm considers adding the label l of a non-simple path to $\mathcal{T}$, there either exists a label with the same costs or for each scenario ξ there exists a label l^ξ dominating l. Hence the algorithm stops after finitely many iterations and finds only simple paths.

In Algorithm 2, for every path p from the source a label l with $z(p) = z(l)$ is constructed, if it does not contain a subpath that is dominated in every scenario. From Lemma 10 it follows that no subpath of a flimsily robust efficient path is dominated in every scenario. Analogous to the proof of Theorem 8, we conclude that for each flimsily robust efficient path p a label l with $z(p) = z(l)$ is found, whereas all labels representing paths which are not flimsily robust efficient are deleted during the algorithm. ∎

An alternative approach to finding a complete set of flimsily robust efficient solutions is presented in Algorithm 3. For each scenario, we use Algorithm 1' to find solutions, which are efficient w.r.t. this scenario. Note that the dominance relation used when applying Algorithm 1' to the subproblems only depends on one scenario. However, when comparing the costs of two labels in Line 6 we only consider them equal if they are equal for each scenario. Therefore, the union of the obtained solution sets is a complete set of flimsily robust efficient solutions. To ensure that Algorithm 3 terminates, we use the same requirement as in Theorem 12: Each cycle C in G has to satisfy either $\forall \xi \in \mathcal{U} : z(C, \xi) = 0$ or $\forall \xi \in \mathcal{U} : 0 \leq z(C, \xi)$.

Algorithm 3 Repeated label correcting algorithm to find flimsily robust efficient solutions

Input: an instance $I = (\mathcal{G}, \mathcal{U}, c, s, t)$ of the multi-objective uncertain shortest path problem
Output: label set $\mathcal{L}$, of which the labels at t represent a complete set of flimsily robust efficient solutions of (*MOUSP*)
1: $\mathcal{L} := \emptyset$
2: **for all** $i = 1, \ldots, r$ **do**
3:   $\mathcal{L}^{\xi_i}$:= output of Algorithm 1' with the relation $(Y^1, Y^2) \in R \Leftrightarrow Y^1_{(\cdot,i)} \leq Y^2_{(\cdot,i)}$
4:   $\mathcal{L} := \mathcal{L} \cup \mathcal{L}^{\xi_i}$

https://doi.org/10.1002/net.21815

4.3 | Highly robust efficiency

Recall that a solution is highly robust efficient if it is efficient for each scenario. We show that for highly robust efficiency Condition 1 is satisfied, but that there exists no binary relation with Property 2(b) as required in Condition 2. However, every highly robust efficient solution is flimsily robust efficient as well. We give an algorithm to find a complete set of highly robust efficient solutions which filters the labels obtained by Algorithm 2. Afterwards, we describe an alternative approach in which we apply Algorithm 1' r times and intersect the obtained solution sets.

Lemma 13. *Let q be a highly robust efficient path for an instance $(G, \mathcal{U}, c, s, t)$ of (MOUSP). Then for every node v in q the subpath $q_{s,v}$ is highly robust efficient for the instance $(G, \mathcal{U}, c, s, v)$, that is, Condition 1 is satisfied.*

Proof. Let v be any node in q. Assume that $q_{s,v}$ is not highly robust efficient for $(G, \mathcal{U}, c, s, v)$. Then there exists a path q' from s to v, which dominates $q_{s,v}$ under at least one scenario $\xi \in \mathcal{U}$. It follows that

$$z(q', \xi) \leq z(q_{s,v}, \xi) \Rightarrow z(q', \xi) + z(q_{v,t}, \xi) \leq z(q_{s,v}, \xi) + z(q_{v,t}, \xi) = z(q, \xi).$$

This contradicts q being highly robust efficient. ∎

Lemma 14. *For highly robust efficiency, there does not exist a relation with Property 2(b), even for only two objectives and two scenarios.*

Proof. Consider the following instance of (*MOUSP*) for $k = r = 2$ with two paths q_1 and q_2 with the following cost matrices:

$$z(\mathcal{Q}^t) = \left\{ Y^1 := \begin{pmatrix} 0 & 5 \\ 0 & 5 \end{pmatrix}, Y^2 := \begin{pmatrix} 1 & 4 \\ 1 & 4 \end{pmatrix} \right\}.$$

Then, none of the two paths in $\mathcal{Q}^t$ is highly robust efficient, because $Y^1_{(\cdot,1)} \leq Y^2_{(\cdot,1)}$ and $Y^2_{(\cdot,2)} \leq Y^1_{(\cdot,2)}$, but both are not dominated by any highly robust efficient path. We conclude that the concept of highly robust efficiency does not have the domination property, hence Property 2(b) cannot hold for any binary relation. ∎

We remark that also Properties 2(a) and 2(c) cannot hold at the same time for the concept of highly robust efficiency.

Without a suitable dominance relation, we cannot use Algorithm 1' to find highly robust efficient solutions. However, since every highly robust efficient solution is also flimsily robust efficient, we can instead compute a complete set of flimsily robust efficient solutions and filter out the highly robust efficient solutions. This can be done efficiently with the help of the additional vectors $x(l)$, which we already introduced for Algorithm 2: At the end of Algorithm 2, a label l is highly robust efficient if $x(l) = (0, \ldots, 0)$. This leads to Algorithm 4.

Algorithm 4 Extended label correcting algorithm to find highly robust efficient solutions

Input: an instance $I = (\mathcal{G}, \mathcal{U}, c, s, t)$ of the multi-objective uncertain shortest path problem
Output: label set $\mathcal{L}$, of which the labels at t represent a complete set of highly robust efficient solutions of (*MOUSP*)
1: $\mathcal{L}$:= output of Algorithm 2
2: **for all** $l \in \mathcal{L}$ **do**
3: **if** $x(l) \neq (0, \ldots, 0)$ **then**
4: $\mathcal{L} := \mathcal{L} \setminus \{l\}$

Theorem 15. *If for each cycle C in G either $z(C, \xi_i) = 0 \, \forall i = 1, \ldots, r$ or $0 \leq z(C, \xi_i) \, \forall i = 1, \ldots, r$, then the output label set $\mathcal{L}$ of Algorithm 4 represents a complete set of highly robust efficient solutions.*

Proof. The statement follows directly from Theorem 12 and the fact that every highly robust efficient solution is flimsily robust efficient. ∎

https://doi.org/10.1002/net.21815

Similar to Algorithm 3 for finding flimsily robust efficient solutions, an alternative approach for finding highly robust efficient solutions is given in Algorithm 5: For each scenario, we use Algorithm 1' to find efficient solutions w.r.t. this scenario. Then we intersect the obtained solution sets. Here again, when applying Algorithm 1' to the subproblems, the dominance relation only depends on one scenario. However, when comparing the costs of two labels in Line 6 of Algorithm 1', we only consider them equal if they are equal for each scenario, in order to obtain a complete set of highly robust efficient solutions in the end. Hence, Algorithm 5 finds a complete set of highly robust efficient solutions for instances where each cycle C in G satisfies either $\forall \xi \in \mathcal{U} : z(C, \xi) = 0$ or $\forall \xi \in \mathcal{U} : 0 \leq z(C, \xi)$.

Algorithm 5 Repeated label correcting algorithm to find highly robust efficient solutions

Input: an instance $I = (\mathcal{G}, \mathcal{U}, c, s, t)$ of the multi-objective uncertain shortest path problem
Output: label set $\mathcal{L}$, of which the labels at t represent a complete set of highly robust efficient solutions of ($MOUSP$)
1: $\mathcal{L} := \emptyset$
2: **for all** $\xi \in \mathcal{U}$ **do**
3: $\mathcal{L}^{\xi_i}$:= output of Algorithm 1' with the relation $(Y^1, Y^2) \in R \Leftrightarrow Y^1_{(\cdot,i)} \leq Y^2_{(\cdot,i)}$
4: $\mathcal{L} := \mathcal{L} \cap \mathcal{L}^{\xi_i}$

4.4 | Point-based and set-based minmax robust efficiency

We show that point-based and set-based minmax robust efficiency both satisfies Condition 2 for using Algorithm 1', but not Condition 1. To be able to nevertheless use a label correcting approach, we propose to use several label sets at each node. This idea was first introduced for single-objective minmax robust shortest path problems in [42].

We first show that both concepts for robust efficiency satisfy Condition 2 by defining a relation for each of the concepts with Properties 2(a)-2(c). Recall that a solution is point-based minmax robust efficient if it is efficient for the deterministic multi-objective problem

$$(SP_{\max}) \quad \min_{q \in Q} \bar{z}(q) := \begin{pmatrix} \max_{\xi \in \mathcal{U}} z_1(q, \xi) \\ \vdots \\ \max_{\xi \in \mathcal{U}} z_k(q, \xi) \end{pmatrix}.$$

($SP_{\max}$) is not a classical multi-objective robust shortest path problem, because suitable edge costs are not known in advance. Therefore, it cannot simply be solved with a deterministic multi-objective labeling algorithm. However, by identifying $z(q)$ with $\bar{z}(q)$, the $\leq$-relation on $\mathbb{R}^k$ induces a binary relation $R^{\text{point}} \subseteq \mathbb{R}^{k \times r} \times \mathbb{R}^{k \times r}$, which is defined as

$$(Y, Y') \in R^{\text{point}} \Leftrightarrow \begin{pmatrix} \max_{i=1,\dots,r} Y_{1,i} \\ \vdots \\ \max_{i=1,\dots,r} Y_{k,i} \end{pmatrix} \leq \begin{pmatrix} \max_{i=1,\dots,r} Y'_{1,i} \\ \vdots \\ \max_{i=1,\dots,r} Y'_{k,i} \end{pmatrix}.$$

It is easy to check that this relation has the properties required in Condition 2. Now, we consider set-based minmax robust efficient solutions: A path $q \in \mathcal{Q}^t$ is set-based minmax robust efficient if there is no solution $q' \in \mathcal{Q}^t$ with $z_{\mathcal{U}}(q') \subseteq z_{\mathcal{U}}(q) - \mathbb{R}^k_{\geq}$. This definition directly leads to a suitable binary relation on $\mathbb{R}^{k \times r}$: Given $k, r \in \mathbb{N}$ we construct the binary relation $R^{\text{set}} \subseteq \mathbb{R}^{k \times r} \times \mathbb{R}^{k \times r}$:

$$(Y, Y') \in R^{\text{set}} \Leftrightarrow \bigcup_{i=1,\dots,r} \left\{ Y_{(\cdot,i)} \right\} \subseteq \bigcup_{i=1,\dots,r} \left\{ Y'_{(\cdot,i)} \right\} - \mathbb{R}^k_{\geq}.$$

Again, it can be checked that this relation fulfills Condition 2.

For $k = 1$, both point-based and set-based minmax robust efficiency reduce to the single-objective concept of minmax robustness. The single-objective minmax robust shortest path problem is already NP-hard [32, 42]. Efficient labeling and dynamic programming algorithms cannot be used, because Bellman's principle of optimality is not satisfied.

https://doi.org/10.1002/net.21815

In [42] a pseudo-polynomial algorithm for the single-objective minmax robust shortest path problem with positive integer edge lengths is given. Instead of a single label at each node v, for each possible cost of the part of the path that has not been looked at yet, a label is saved at v.

In order to find a complete set of [set-based/point-based] minmax robust efficient solutions, we transfer this idea to our label correcting algorithm by adding a *prediction matrix* as a fourth component to each label: A label $l = (v, z(l), l', A)$ now consists of a node v, a cost matrix $z(l)$, a predecessor label l' as before, and a prediction matrix $A \in \mathbb{Z}^{k\times r}$. We also define a function a with $a(l) := A$, assigning the prediction matrix to label l. A path from s to v can be represented by several labels with different prediction matrices. The prediction matrix contains the assumed costs for continuing the path from v to t.

In the beginning of the algorithm, component-wise upper and lower bounds $A_{i,j}^{\min}$ and $A_{i,j}^{\max}$ for the cost of a simple path in G are computed. For example, one obtains suitable bounds by

$$A_{i,j}^{\min} := \sum_{e\in E} \min\left\{0, c_i(e, \xi_j)\right\}, \quad A_{i,j}^{\max} := \sum_{e\in E} \max\left\{0, c_i(e, \xi_j)\right\}.$$

With $A \leqq A'$ we denote that matrix A is component-wise smaller or equal to matrix A'.

Algorithm 6 is correct for instances with integer edge costs which are conservative w.r.t. [R^{point} /R^{set}]. However, it can easily be adjusted to rational edge costs by allowing $A \in \mathbb{Q}^{k\times r}$ and adjusting the step length by which $A_{i,j}$ is increased in Lines 3 to 6.

Algorithm 6 Extended label correcting algorithm to find [set-based/point-based] minmax robust efficient solutions

Input: an instance $I = (G, \mathcal{U}, c, s, t)$ of the multi-objective uncertain shortest path problem (*MOUSP*) with $c(e) \in \mathbb{Z}^{k\times r}\ \forall\, e \in E$

Output: label set $\mathcal{L}$, of which the labels at t with prediction matrix 0 represent a complete set of [point-based/set-based] minmax robust efficient solutions of (*MOUSP*)

1: Compute lower and upper bounds $A_{i,j}^{\min}$ and $A_{i,j}^{\max}$ for the path cost components.
2: $\mathcal{T} := \emptyset, \mathcal{L} := \emptyset$
3: **for** $A_{1,1} = A_{1,1}^{\min}, \dots, A_{1,1}^{\max}$ **do**
4: **for** $A_{1,2} = A_{1,2}^{\min}, \dots, A_{1,2}^{\max}$ **do**
5: ...
6: **for** $A_{k,r} = A_{k,r}^{\min}, \dots, A_{k,r}^{\max}$ **do**
7: Set $l_0^A := (s, 0, 0, A), \mathcal{T} := \mathcal{T} \cup \left\{l_0^A\right\}$
8: **while** $\mathcal{T} \neq \emptyset$ **do**
9: Choose a label $l' \in \mathcal{T}$ at any node v' and set $\mathcal{T} := \mathcal{T} \setminus \{l'\}, \mathcal{L} := \mathcal{L} \cup \{l'\}$.
10: **for all** outgoing edges $e = (v', v)$ of v' **do**
11: Set $l := (v, z(l') + c(e), l', a(l') - c(e))$.
12: **if** $A^{\min} \leqq a(l) \leqq A^{\max}$ and $\nexists\, \tilde{l} = (v, z(\tilde{l}), \tilde{l}', a(l)) \in \mathcal{T} \cup \mathcal{L}$ with $z(\tilde{l}) = z(l)$ or $(z(\tilde{l}) + a(l), z(l) + a(l)) \in R$ **then**
13: $\mathcal{T} := \mathcal{T} \cup \{l\}$
14: **for all** labels $\tilde{l} = (v, z(\tilde{l}), \tilde{l}', a(l)) \in \mathcal{T}$ with $(z(l) + a(l), z(\tilde{l}) + a(l)) \in R$ **do**
15: $\mathcal{T} := \mathcal{T} \setminus \left\{\tilde{l}\right\}$
16: **for all** labels $\tilde{l} = (v, z(\tilde{l}), \tilde{l}', a(l)) \in \mathcal{L}$ with $(z(l) + a(l), z(\tilde{l}) + a(l)) \in R$ **do**
17: $\mathcal{L} := \mathcal{L} \setminus \left\{\tilde{l}\right\}$

Theorem 16. *Let R be the relation [R^{point}/R^{set}]. Let the instance $I = (G, \mathcal{U}, c, s, t)$ be conservative w.r.t. R and $c(e) \in \mathbb{Z}^{k\times r}\ \forall\, e \in E$. Then the output of Algorithm 6 is a complete set of [point-based/set-based] minmax robust efficient solutions.*

Proof. We first show that Algorithm 6 stops after finitely many iterations. We then show that $q \in \mathcal{Q}^t$ is [point-based/set-based] minmax robust efficient $\Leftrightarrow$ at the end of Algorithm 6, there is a label $l \in \mathcal{L}$ at node t with cost $z(l) = z(q)$ and prediction matrix $a(l) = 0$.

First note that in contrast to Algorithm 1 and Algorithm 2, in Algorithm 6 labels corresponding to non-simple paths are not immediately sorted out in Line 12, since Line 12 only compares labels having the same prediction matrix. However, since there are only $\hat{m} := \prod_{i=1,\dots,k,j=1,\dots,r} \left(A_{i,j}^{\max} - A_{i,j}^{\min} + 1\right)$ different prediction matrices, no path

https://doi.org/10.1002/net.21815

for which a label is added to $\mathcal{L}$ contains a node v more than $\hat{m}$ times: A path p containing v more than $\hat{m}$ times has at least $\hat{m}+1$ subpaths ending in v (including p itself). Hence, at least two of the corresponding labels have the same prediction matrix. However, as soon as a label l' in v is created with the same prediction matrix as a predecessor label l in v, we have

$$z(l) = z(l') \text{ or } (\mathrm{z(l) + a(l), z(l') + a(l')}) \in \mathrm{R}$$

since the instance is conservative, and l' is discarded in Line 12. We now show that $q \in \mathcal{Q}^t$ is [point-based/set-based] minmax robust efficient $\Leftrightarrow$ at the end of the algorithm, there is a label $l \in \mathcal{L}$ at node t with cost $z(l) = z(q)$ and prediction matrix $a(l) = 0$.

$\Rightarrow$: Let q be a [point-based/set-based] minmax robust efficient solution. Without loss of generality we can assume that q is a simple path: Because the instance is conservative w.r.t. R, for any non-simple path q there either exists an equivalent simple path or q is not [point-based/set-based] minmax robust efficient.

Let l be the first label at t added to $\mathcal{T}$ with cost $z(l) = z(q)$ and $a(l) = 0$. Then $l \in \mathcal{L}$ at the end of the algorithm, because there exists no $q' \in \mathcal{Q}^t$ with $(z(q'), z(q)) \in R$. It remains to show that a label with cost $z(l) = z(q)$ and $a(l) = 0$ is added to $\mathcal{T}$. We show by induction that for each node v on q, a label with cost $z(q_{s,v})$ and prediction matrix $z(q_{v,t})$ is added to $\mathcal{T}$ during the algorithm.

In Line 7, a label at node s with length 0 and prediction matrix $z(q)$ is added to $\mathcal{T}$, since $A^{\min} \leqq z(q) \leqq A^{\max}$. Let (v', v) be an edge in q. Assume that a label l' at v' with $z(l') = z(q_{s,v'})$ and $a(l') = z(q_{v',t})$ is added to $\mathcal{T}$ during the algorithm. Since q is [point-based/set-based] minmax robust efficient, there is no path q' with $(z(q') + z(q_{v',t}), z(q_{s,v'}) + z(q_{v',t})) \in R$. Hence, l' is removed from $\mathcal{T}$ and added to $\mathcal{L}$ in some iteration of Line 9. Then, in Line 11 a label l with $z(l) = z(q_{s,v'}) + c(v', v) = z(q_{s,v})$ and $a(l) = z(q_{v',t}) - c(v', v) = z(q_{v,t})$ is created. Since q is [point-based/set-based] minmax robust efficient, there is no path q' with $(z(q') + z(q_{v',t}), z(q_{s,v'}) + z(q_{v',t})) \in R$ and l is added to $\mathcal{T}$, unless there already is a label in $\mathcal{T} \cup \mathcal{L}$ with the same cost and prediction matrix.

We conclude that for each node v on q, a label with cost $z(q_{s,v})$ and prediction matrix $z(q_{v,t})$ is added to $\mathcal{T}$ during the algorithm, in particular for $v = t$.

$\Leftarrow$: The dominance checks in Lines 12–17 guarantee that for any two labels l, l' in $\mathcal{L}$ we have

$$(z(l) + a(l), z(l') + a(l')) \notin R,$$

thus in particular for our output labels (with $a(l) = a(l') = 0$)

$$(z(l), z(l')) \notin R,$$

no two paths in the output dominate each other.

■

Note that this algorithm also only returns labels representing simple paths: If a non-simple path p is not dominated, the cost of all its cycles is 0 and the label representing the respective simple path with prediction matrix 0 was constructed earlier than the label representing p.

4.5 | Summary

Table 1 summarizes which properties of the two conditions given in Section 3 are satisfied for each of the considered concepts of robust efficiency and which algorithms can be used to find a complete set of robust efficient solutions. All presented algorithms are pseudo-polynomial for a fixed number of objectives and scenarios and integer edge costs: Carefully counting the steps shows that for polynomially bounded integer edge costs the algorithms run in polynomial time. This cannot be expected if the number of scenarios is unbounded, since the single-objective minmax robust shortest path problem with integer edge costs is then already strongly NP-hard [42].

https://doi.org/10.1002/net.21815

TABLE 1 Summary of which conditions are satisfied for which concept of robust efficiency and which algorithms can be used to solve (*MORSP*)

Concept of robust efficiency	Condition 1	Condition 2	Algorithms
Multi-scenario efficiency	Yes	Yes	1
Flimsily robust efficiency	Yes	No	2,3
Highly robust efficiency	Yes	No	4,5
Point-based minmax robust efficiency	No	Yes	6
Set-based minmax robust efficiency	No	Yes	6

5 | EXPERIMENTS

In the previous section we developed several algorithms for finding robust efficient solutions. These can be classified into two groups:

- Extended labeling algorithms: Algorithms that use an extension of Algorithm 1' based on the Conditions 1 and 2 we introduced in Section 3. These are Algorithms 2, 4, and 6 for flimsily, highly, and point-based/set-based minmax robust efficiency.
- Repeated labeling algorithms: Algorithms that rely on repeated application, for every scenario, of Algorithm 1'. These are Algorithms 3 and 5 for flimsily and highly robust efficiency

The main goal of this section is to compare these two classes of algorithms. Since we have algorithms from both classes for the two concepts of flimsily and highly robust efficient solutions we take these as basis for our experiments, that is, the following four algorithms presented in this article are tested and compared in detail:

- EL-Flimsily is Algorithm 2, the extended label correcting algorithm to find flimsily robust efficient paths.
- RL-Flimsily is Algorithm 3, where Algorithm 1' is applied r times to find flimsily robust efficient paths.
- EL-Highly is Algorithm 4, which applies Algorithm 2 (EL-Flimsily) and identifies highly robust efficient solutions from the output.
- RL-Highly is Algorithm 5, where Algorithm 1' is applied r times to find highly robust efficient paths.

In addition, we also present some results showing particularities of the extended labeling algorithms for finding point-based and set-based minmax robust efficient solutions:

- EL-PB is Algorithm 6 with dominance relation R^{point}, the extended label correcting algorithm to find point-based minmax robust efficient paths.
- EL-SB is Algorithm 6 with dominance relation R^{set}, the extended label correcting algorithm to find set-based minmax robust efficient paths.

Since our test instances have positive edge lengths, we set the lower bounds $A_{i,j}^{\min}$ needed for EL-PB and EL-SB to 0. Further, we calculate the upper bounds $A_{i,j}^{\max}$ as the sum of the $|V| - 1$ largest costs for each objective i and scenario j.

All algorithms were implemented in C++, compiled with gcc version 5.4.0, and run under Ubuntu 16.04.2 on a laptop with 3GHz processor and 16GB RAM. Results are analyzed and plots and tables are generated in the statistical computing environment R [36].

5.1 | Test instances

We test the presented algorithms based on two types of network instances, grid networks and so-called NetMaker networks.

5.1.1 | Grid networks

Grid networks are introduced in [11, 37], where nodes are arranged in a rectangular grid of height h and width w. The start node s and end node t are outside the grid, namely on the left and right, with edges connecting them to all left-most and right-most

https://doi.org/10.1002/net.21815

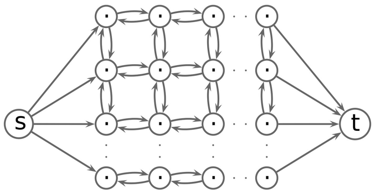


FIGURE 2 Structure of grid networks

nodes, respectively, as shown in Figure 2. The (integer) edge cost components are randomly chosen from a discrete uniform distribution between 1 and a given upper bound c. We construct one set of *random* grid network instances where the costs for all scenarios are chosen randomly. For the other set of *correlated* grid network instances the cost vector of scenario ξ_1 is randomly generated, and the other cost vectors $c(e, \xi)$ are generated based on $c(e, \xi_1)$, where costs are now randomly generated such that $c(e, \xi) \in \{\max\{1, c(e, \xi_1) - 3\}, \dots, c(e, \xi_1) + 3\}$.

5.1.2 | NetMaker networks

So-called NetMaker networks were first introduced by [40] for testing a bi-objective shortest path algorithm, and also used by others [37]. A random Hamiltonian cycle ensures the network is connected. Other edges (v, v') are randomly generated for each node v. A random number of edges with tail node v are generated where the number of such edges lies in the interval $\{e_{\min}, e_{\min} + 1, \dots, e_{\max}\}$ with equal probability. NetMaker also limits how far these edges can reach: Assuming all nodes are numbered $\{1, 2, 3, \dots, |V|\}$, with 1 being the start node and $|V|$ the target node, any edge (v, v') with tail node v can only reach a node $v' \in \left\{v - \left\lceil \frac{I}{2} \right\rceil, \dots, v-1, v, v+1, \dots, v + \left\lceil \frac{I}{2} \right\rceil\right\}$. This prevents paths from s to $|V|$ with very few edges, which then would easily dominate all other paths. In the following $I = 10$ is chosen.

In the original bi-objective instances [37, 40], for each edge it is first randomly determined which interval edge costs fall into:

1. $c_1(e) \in \{1, 2, \dots, 33\}$ and $c_2(e) \in \{67, 68, \dots, 100\}$, or
2. $c_1(e) \in \{67, 68, \dots, 100\}$ and $c_2(e) \in \{1, 2, \dots, 33\}$.

The actual edge cost is then randomly chosen from the respective set, with uniform distribution. To generate instances with $k = 3$, for each edge, we randomly allocate exactly one of the three cost intervals $\{1, 2, \dots, 33\}$, $\{34, 35, \dots, 66\}$, or $\{67, 68, \dots, 100\}$ to each edge cost component, and randomly select the actual cost value from the respective interval.

We generate NetMaker network instances with *random* scenarios. For any edge e and cost component k, all scenarios' costs $c_k(e, \xi)$ will be randomly chosen from the same interval associated with k and e. *Correlated* NetMaker instances are constructed as for grid networks by randomly generating the cost vector $c(e, \xi_1)$ according to scenario ξ_1, and generating the others such that $c(e, \xi) \in \{\max\{1, c(e, \xi_1) - 3\}, \dots, c(e, \xi_1) + 3\}$. The costs of edges, for all cost components and scenarios, on the Hamiltonian cycle are chosen like all other edge costs, and multiplied by a factor of 10 to penalize their use. In this aspect our instance generation may differ from [40].

5.2 | Finding flimsily and highly robust efficient solutions

This section analyses solution numbers, difficulty of problem instances and runtimes of the different algorithms introduced for finding flimsily and highly robust efficient solutions.

5.2.1 | Computational setup

We consider instances based on grid networks with two or three objectives ($k = 2, 3$) with each combination of the following parameters:

- grid height $h = 10, 20, 30, 40$,
- grid width $w = 10, 20, 30, 40$,

https://doi.org/10.1002/net.21815

- number of scenarios $r = 2, 4, 6, 8$, and
- costs chosen from $\{1, 2, \ldots, c\}$ with $c = 10, 100$.

The edge cost range used in [11, 37], $\{1, \ldots, 10\}$, is therefore considered in our setting. We furthermore test a smaller range of instances with four objectives ($k = 4$) and the following parameters:

- grid height $h = 10, 20, 30$,
- grid width $w = 10, 20, 30$,
- number of scenarios $r = 2, 4$, and
- costs chosen from $\{1, 2, \ldots, c\}$ with $c = 10, 100$.

We consider instances based on NetMaker networks with each combination of the following parameters:

- number of objectives $k = 2, 3$
- number of nodes $n = 101, 201, 401, 801, 1201$, also $n = 1601$ only for $k = 2$
- number of outgoing edges for each node in $\{1, 2, \ldots, 7\}$, that is, $e_{\min} = 1, e_{\max} = 7$.
- This ensures that, on average, there are 4–5 outgoing edges for each node. This leads to similar network density in grid and NetMaker networks.
- number of scenarios $r = 2, 4, 6, 8$.

Tables A1–A16 in the appendix list $|V|, |E|$ and the choice of parameters h, w, r, c for each grid instance; similarly NetMaker instance parameters are listed in Tables A17–A24. Runtime (in seconds) is recorded for each algorithm in the tables. When runtime exceeds 1 hour, runs were not completed and the runtime is shown in the tables as > 3600.00. The tables also list the number of solutions found for each instance, where the column "sols" refers to the number of obtained flimsily and highly robust efficient solutions, respectively.

For most experiments a single instance was generated for each set of parameters. Since costs in grid networks, as well as edges and costs in NetMaker networks, are randomly chosen, instances for the same set of parameters can vary. For NetMaker instances we analyze the results over repeated runs (20 for $k = 2$ and 10 for $k = 3$) for each set of problem parameters. Hence, minimum, maximum and averages are reported in Tables A17–A24, and plots in the following subsections show average results and error bars (one standard deviation), where applicable. For grid networks, where only the edge costs, not the network structure itself, are variable, we investigate the variability of runtimes and numbers of solutions for $k = 2$ objectives on 20 instances for each parameter set (see Section 5.2.2). Results for $k = 2$ in Tables A1–A4 and A9–A12 also report minimum, maximum and average, and plots are based on average results, with error bars where applicable.

5.2.2 | Comparison of extended and repeated labeling algorithms

Tables A1–A24 show that the runtimes of the extended labeling algorithms EL-Flimsily and EL-Highly are in general similar for each instance, which is expected as they both apply Algorithm 2. Similarly, runtimes of the repeated labeling algorithms RL-Flimsily and RL-Highly are similar as they also both apply Algorithm 1' r times. When runtimes differ this is due to the complexity of the filtering process to identify all flimsily or highly robust efficient solutions. Hence, we will illustrate all results about runtimes only for flimsily robust efficiency. The same trends can be observed for highly robust efficiency as well, if not stated otherwise.

Figures 3 and 4 show runtimes of both classes of algorithms for finding flimsily robust efficient solutions in grid networks with correlated and random edge costs. The horizontal axis shows network height and width of the instances, and the two different types of algorithms are shown as circles and triangles with points slightly offset to make them easier to compare. The white background color indicates results for $c = 10$, and gray background for $c = 100$. For $k = 2$ average runtimes are shown by the marker with error bars indicating one standard deviation. Furthermore, the number of scenarios in an instance is color-coded.

We observe that it is faster to solve Algorithm 1' r times, as in the repeated labeling algorithms RL-Flimsily and RL-Highly, than to tackle the full problem with the extended labeling algorithms EL-Flimsily and EL-Highly, respectively. This is due to the increased complexity of the algorithms as the additional vector x has to be maintained to correctly determine dominance of flimsily robust efficient labels. Discarding a label because it is dominated may only be possible later during the algorithm as a

https://doi.org/10.1002/net.21815

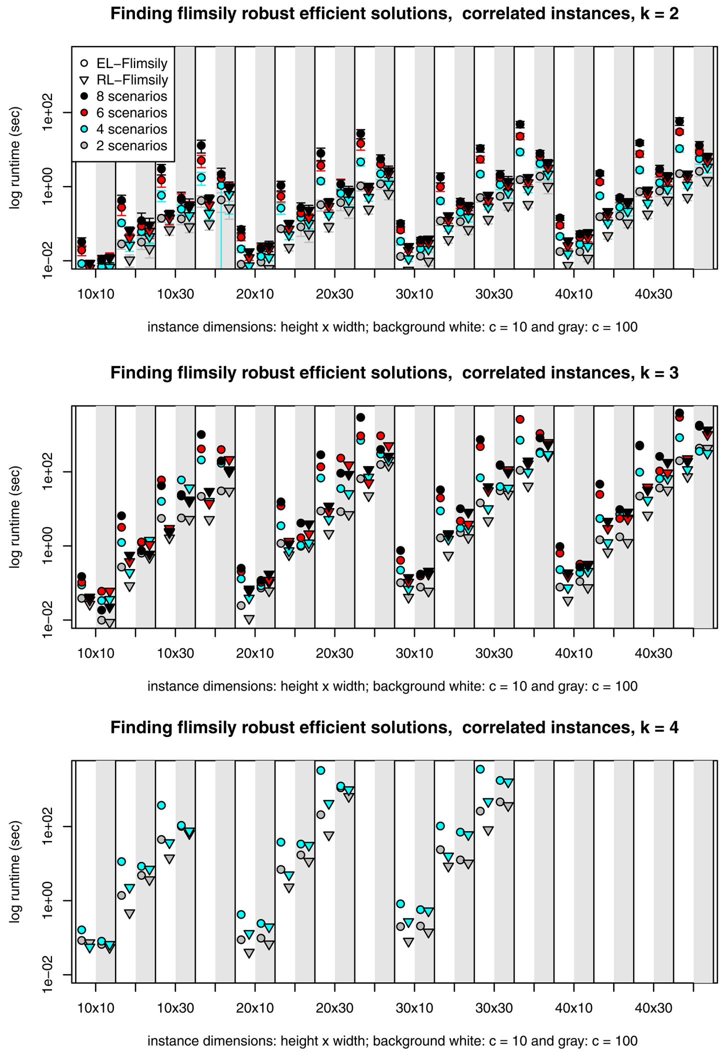


FIGURE 3 Runtimes for finding flimsily robust efficient solutions of correlated grid instances with $k = 2$ (top), $k = 3$ (center), $k = 4$ (bottom), only showing runtimes exceeding 0.01 seconds. For $k = 2$ averages are shown with error bars indicating one standard deviation [Color figure can be viewed at wileyonlinelibrary.com]

label can only be discarded once it is dominated in all scenarios, when $x = (1, 1, \dots, 1)$. Hence, before it is confirmed that a label cannot be flimsily robust efficient, it may have been extended to many other nodes. The advantage of solving Algorithm 1' r times, as in RL-Flimsily and RL-Highly, is that the subproblems have fewer labels at the nodes as dominance can be established earlier, namely as soon as a label is dominated in the current scenario. This means that labels are less often unnecessarily carried forward by the algorithms.

For random instances (Figure 3) runtimes of RL-Flimsily and EL-Flimsily increase, when the maximum cost increases from $c = 10$ to $c = 100$. For correlated instances, runtimes of RL-Flimsily increase, whereas runtimes of EL-Flimsily decrease when the maximum cost increases. The number of flimsily robust efficient solutions tends to decrease (see corresponding tables), and the runtime of EL-Flimsily benefits from this. Finally, the repeated runs for the same set of instance parameters with $k = 2$ show

https://doi.org/10.1002/net.21815

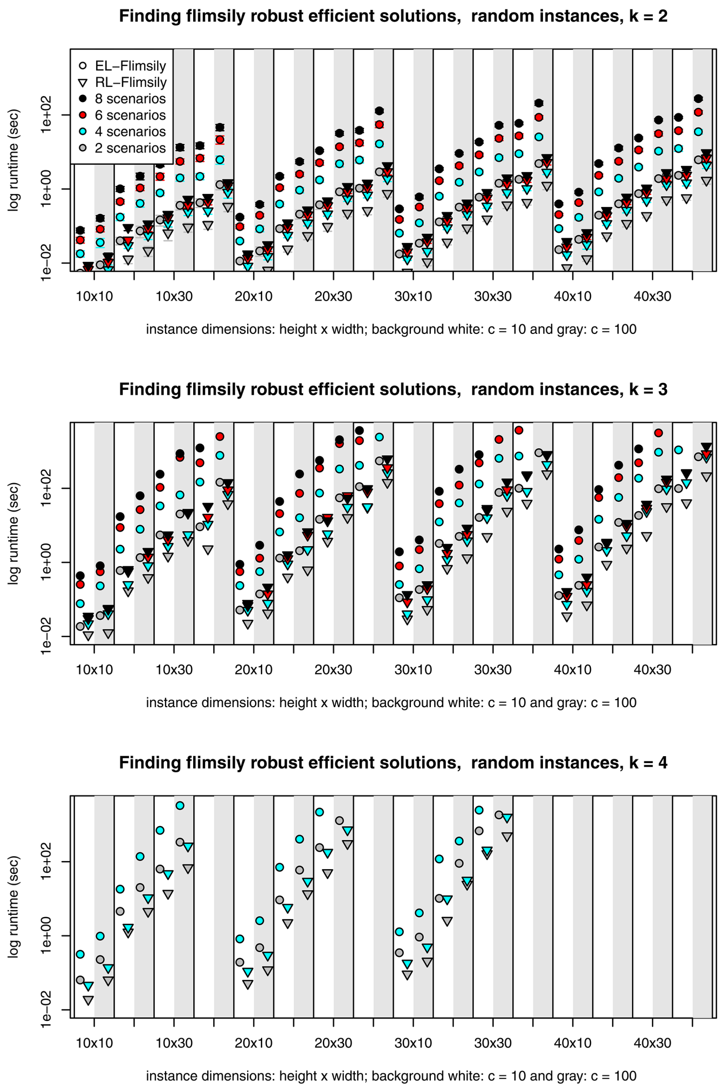


FIGURE 4 Runtimes for finding flimsily robust efficient solutions of random grid instances with $k = 2$ (top), $k = 3$ (center), $k = 4$ (bottom), only showing runtimes exceeding 0.01 seconds. For $k = 2$ averages are shown with error bars indicating one standard deviation [Color figure can be viewed at wileyonlinelibrary.com]

that instances are of varying difficulty in terms of number of solutions and runtime, as expected. For instances with random scenarios the effect was minor; that is runtimes for one set of parameters generally do not overlap with those for a different set of parameters. While runtimes for similar sets of instance parameters can overlap for correlated instances, for example, for 6 and 8 scenarios, this does not tend to occur for parameter values that differ more, for example, 2 and 8 scenarios. Therefore we conclude that general trends observed for grid networks in this section are valid even though experiments were only run for one instance per set of parameters when $k = 3, 4$.

Figure 5 shows runtimes for NetMaker instances. Here, the average runtimes of the two classes of algorithms, indicated by circles and triangles, are shown for networks of different sizes. The error bars indicate one standard deviation. The number of

https://doi.org/10.1002/net.21815

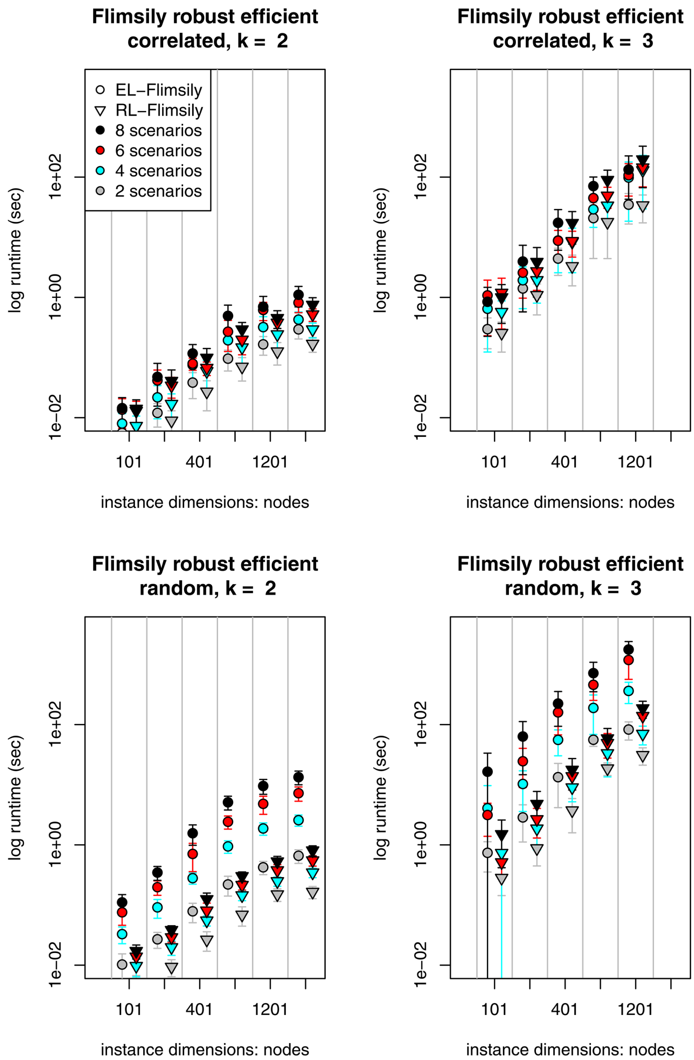


FIGURE 5 Runtimes for finding flimsily robust efficient solutions of NetMaker instances with $k = 2$ (left), $k = 3$ (right), correlated (top) and random (bottom) scenarios, only showing runtimes exceeding 0.01 seconds. Markers indicate average runtimes, and error bars one standard deviation [Color figure can be viewed at wileyonlinelibrary.com]

scenarios is color-coded and the subfigures show instances with $k = 2$ or $k = 3$ objectives. We observe that for this network type, the extended algorithm is sometimes faster than the repeated algorithm, in particular for correlated scenarios and $k = 3$ (Figure 5). This is illustrated in Figure 6, where runtimes of the EL-Flimsily and RL-Flimsily algorithms are plotted for the same set of parameters, and the straight line indicates where runtimes would be equal. It again confirms that for some correlated instances with $k = 3$ EL-Flimsily is faster than RL-Flimsily.

This can be explained by the fact that for correlated scenarios a path is more likely to dominate another in every scenario than for random scenarios. As explained above, in EL-Flimsily, a label that does not represent a flimsily robust efficient path may produce many successor labels until a dominating label is found for each scenario. In instances with correlated scenarios, however, a label is often dominated for all scenarios, as soon as it is dominated for one scenario, hence one dominating label

https://doi.org/10.1002/net.21815

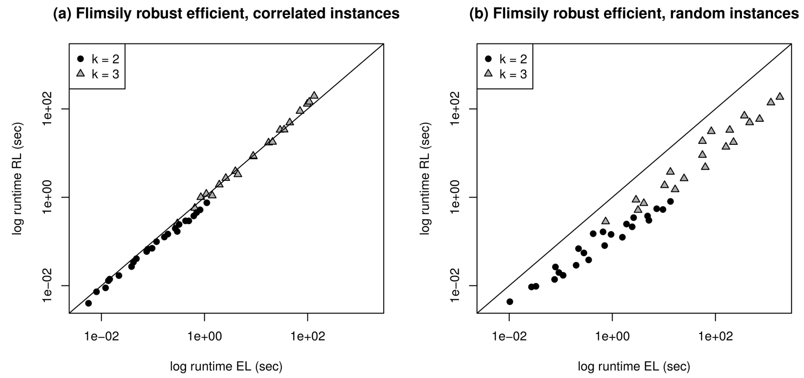


FIGURE 6 Comparing the average runtimes for NetMaker instances for finding flimsily robust efficient solutions for (a) correlated and (b) random instances. The straight line indicates where instances would have equal runtimes for both algorithms

suffices to discard it. The runtime of EL-Flimsily benefits from this, whereas RL-Flimsily needs to repeat the whole labeling procedure r times, even if the costs are identical for all scenarios. This effect can also be observed for grid networks, when comparing the runtime of random and correlated instances in Figures 3 and 4 (in particular for $c = 100$); however, RL-Flimsily is still faster even for correlated grid instances. The difference between runtimes for random and correlated instances is discussed in more detail in Section 5.2.4.

5.2.3 | Runtime with respect to network size and number of scenarios and objectives for both classes

For grid networks, Figures 3 and 4 show how instances become more challenging as the height or width of the problem instance increases. This increase is more significant for increasing width than for increasing height, which is explained in Section 5.2.5.

Comparing the plots for $k = 2, 3, 4$, which all use the same scale for runtime, it is apparent that increasing k significantly increases the runtime. Further, for higher numbers of objectives, the parameters h and w influence runtime more, as can be seen by comparing the difference between runtimes for different network sizes in each of the plots.

In addition, the number of scenarios is color-coded in the figures and illustrates that the runtime of both classes of algorithms mostly increases as the number of scenarios increases. However, this trend is not as clear as for increasing size of networks and number of objectives, as can, for example, be observed for several instances with 6 or 8 scenarios, in particular for correlated instances.

Similarly, Figure 5 shows that also for NetMaker instances increasing the numbers of nodes, objectives and scenarios generally lead to increasing runtimes.

5.2.4 | Differences between correlated and random scenarios

We also analyze differences in runtime and number of solutions for random and correlated instances with the same parameters.

For grid instances, by comparing Figures 3 and 4 (and corresponding tables), one can observe that runtimes for EL-Flimsily tend to be lower for correlated instances than for random instances, in particular for $c = 100$. An explanation for this is given in Section 5.2.2. In Figure 7 we analyze differences in runtime of RL-Flimsily and number of solutions found for random and correlated scenarios. Every point in Figure 7a represents the number of solutions of a grid instance with parameters k, r, h, w with correlated scenarios (horizontal axis) and random scenarios (vertical axis). For $k = 2$ it shows the average number of solutions of all 20 instances with the same set of parameters. It should be noted that, for all instances contributing to the same point in the figure, the instance parameters k, r, h, w are identical, but instances have different randomly generated costs associated with the edges. The straight line indicates where the number of solutions for random and correlated instances is identical. The figures distinguish instances with $c = 10$ (circles) and $c = 100$ (triangles).

Figure 7a shows that the number of flimsily robust efficient solutions found for instances with correlated and random scenarios is often similar but some random instances with $c = 10$ tend to have more solutions than their correlated counterpart (there are more points further above the line than below). For $c = 100$ a clear trend for more solutions in random scenarios can be seen. Runtimes (or average runtimes, for $k = 2$) of RL-Flimsily in Figure 7b tend to be similar for correlated and random

https://doi.org/10.1002/net.21815

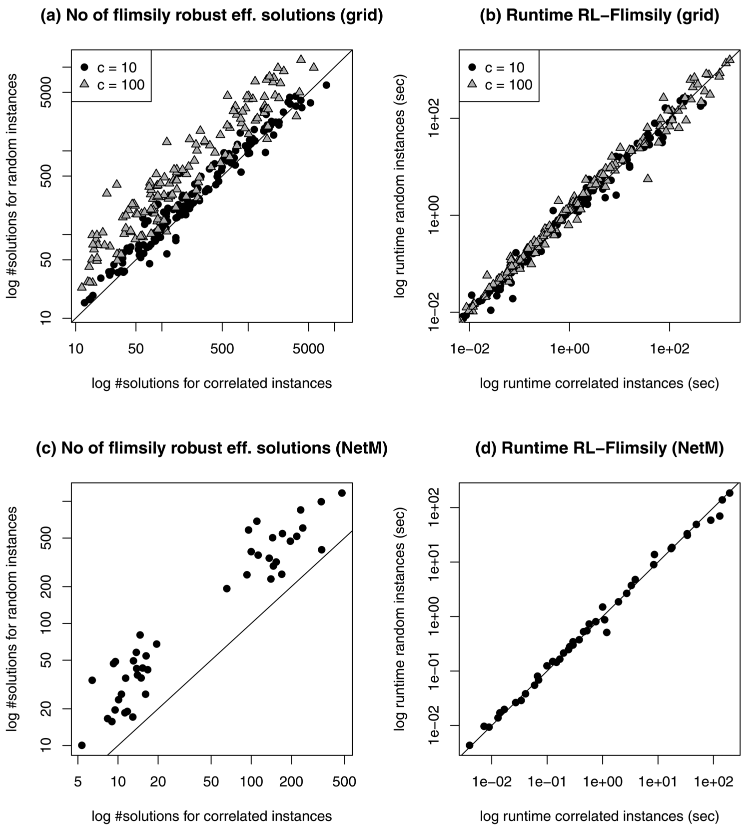


FIGURE 7 Comparing the number of flimsily robust efficient solutions (a) and RL-Flimsily runtimes (b) between grid network instances with correlated and random scenarios (based on averages for $k = 2$); similarly for NetMaker instances in (c) and (d) (also based on averages). The straight line indicates where instances would have equal numbers of solutions and runtimes, respectively

scenarios (points are close to the line), despite more solutions for random scenarios. This is likely due to similar numbers of efficient solutions found for each scenario, which, in the correlated case, are often the same solution, whereas they are more likely to be distinct solutions in the random case.

Similarly, comparing the plots in Figure 5, it is apparent that the runtime of EL-Flimsily tends to be much lower for correlated NetMaker instances than for random NetMaker instances, as explained in Section 5.2.2. We do not observe this for RL-Flimsily. In Figure 7c,d this is investigated in more detail, similar to Figure 7a,b for grid instances. Again, the average results over all instances with the same parameters are shown. Figure 7d shows runtimes of RL-Flimsily, which tend to be similar for random and correlated instances, even though the number of solutions tends to be higher for random instances, as shown in Figure 7c.

5.2.5 | Number of robust efficient solutions

There generally are many flimsily robust efficient solutions, and fewer highly robust efficient solutions.

We note that grid network instances with random scenarios in our experiments do generally not have any highly robust efficient solutions for $k = 2, 3$, see Tables A11–A14, whereas instances with $k = 4$ tend to have a few, mainly for $r = 2$ (Tables A15, A16). For grid network instances with correlated scenarios more highly robust efficient solutions are found, see Tables A3–A9, as a solution that is efficient in one scenario is more likely to also be efficient in another (correlated) scenario. This effect is stronger for $c = 100$, when compared to $c = 10$, leading to more highly robust efficient solutions when $c = 100$. In

https://doi.org/10.1002/net.21815

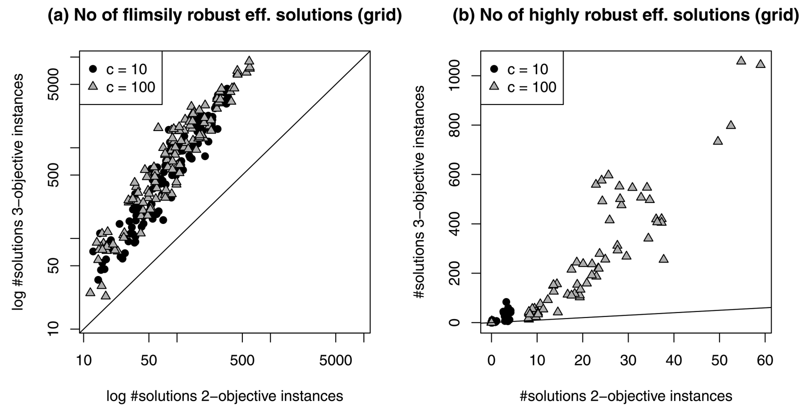


FIGURE 8 Comparing the number of flimsily (a) and highly (b) robust efficient solutions of correlated and random grid network instances with 2 and 3 objectives (both plots are based on averages for $k = 2$). The straight line indicates where instances would have equal numbers of solutions

addition, it can be observed that the number of highly robust efficient solutions increases as the number of objectives increases, and that it tends to be higher for fewer scenarios.

As instance size, number of scenarios, and number of objectives increase, the number of flimsily robust solutions found also increases. Figure 8a shows that instances with the same parameters h, w, r with $k = 2$ objectives (horizontal axis) and $k = 3$ objectives (vertical axis) clearly have more solutions for $k = 3$. Figure 8b illustrates the increase in highly robust efficient solutions found for $k = 3$, again compared to $k = 2$.

Our results also show that problem instances become more challenging as their size increases, that is as h and w increase. On closer inspection wider networks are more challenging than higher networks. For example, instances with $h = 20, w = 30$ have more flimsily robust efficient solutions and longer runtimes than instances with $h = 30, w = 20$. Narrow and high networks tend to have shorter paths and fewer flimsily robust efficient paths as paths tend to dominate each other more. Wide networks, on the other hand, have longer and more flimsily robust efficient paths as there are more possible ways of traversing the network on paths that do not dominate each other.

Random NetMaker instances also tend to have few highly robust efficient solutions, in particular for only two objectives, where often no highly robust efficient solution exists (see Tables A23, A24). Correlated instances, however, tend to have more highly robust efficient solutions, since an efficient path w.r.t. one scenario is much more likely to be efficient w.r.t. the other scenarios as well, if the edge costs in all scenarios are similar. As for grid network instances, NetMaker instances with three objectives generally have more flimsily and highly robust efficient solutions than instances with only two objectives.

5.3 | Finding point-based and set-based minmax robust efficient solutions

EL-PB and EL-SB are, already for small matrices, demanding in terms of runtime and memory usage. A RAM limit of 14 GB did only allow the solution of instances with very small networks. The memory usage and runtime increase rapidly with increasing number of scenarios and/or objectives.

In addition the memory usage and runtimes for instances with the same number of objectives and scenarios and the same network structure differ greatly. An important factor is the number of prediction matrices used in the algorithm, since the number of constructed labels relies heavily on it, which we demonstrate based on a grid network with $h = w = 2$, $k = 2$ and $r = 2$. The integer edge cost components are randomly chosen between 1 and c from a discrete uniform distribution, where c lies between 2 and 12. For each $c \in \{2, \dots, 12\}$, ten random instances are created. Figure 9 shows the runtime of EL-PB in relation to the number of prediction matrices for this 2×2 grid network with $r = 2$ and $k = 2$. The runtimes of EL-SB show the same trend and are omitted here.

From the theory we know that the number of considered prediction matrices depends on the number of objectives and scenarios and on the lower and upper bounds $A_{i,j}^{\min}$ and $A_{i,j}^{\max}$. Hence, in addition to the number of objectives and scenarios, also the lower and upper bounds play a critical role regarding the runtime (and memory usage) of EL-PB and EL-SB. Since we look

https://doi.org/10.1002/net.21815

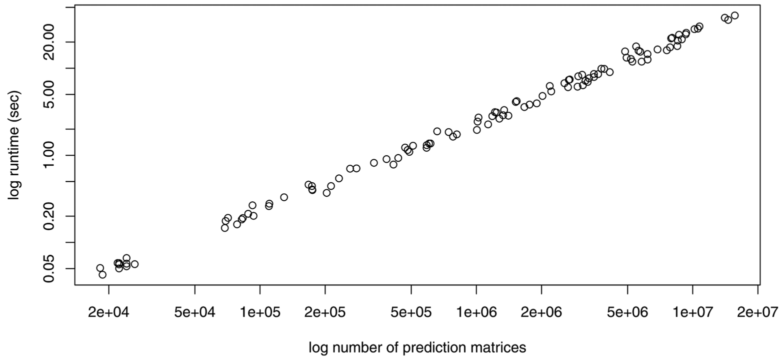


FIGURE 9 Runtime of EL-PB for several instances of edge costs for a grid network with $h = w = 2, k = 2$, and $r = 2$, in relation to the number of prediction matrices produced for this instance during execution of the algorithm

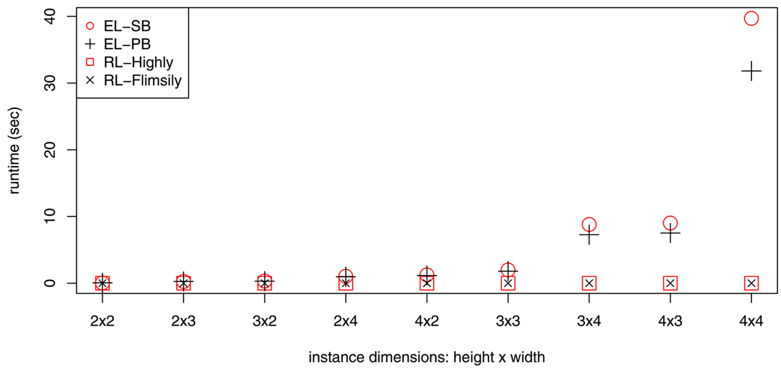


FIGURE 10 Runtime for grid network instances with two objectives and two scenarios (average of 10 randomly generated instances for each network size with cost components in {1,2}) [Color figure can be viewed at wileyonlinelibrary.com]

for the labels with prediction matrix 0 at node t, the lower bounds cannot be chosen higher than 0. However, the upper bounds depend on the $|V| - 1$ maximal edge costs, and thus on the maximal possible edge cost c.

As a consequence, to be able to compare networks of different sizes we consider $k = 2$ objectives, $r = 2$ scenarios and edge costs chosen randomly from $\{1, 2\}$ (uniformly distributed). Figure 10 shows the average runtime of ten random instances each for different sizes of grid networks with width and height between 2 and 4. One can see that the average runtimes of EL-PB and EL-SB increase significantly with the size of the network, even for the small networks considered here. Instances with grid networks of width and height larger than 4 could not be solved due to memory capacities. In comparison, the time needed to find flimsily or highly robust efficient solutions increases much more slowly. Further, the runtime of EL-SB is higher and increases faster than the runtime of EL-PB. This can be explained by the complexity of the comparison procedure: to check whether a pair of label costs is in R^{set} takes more time than to check whether it is in R^{point}.

5.4 | Summary

In summary, it is challenging to identify robust efficient solutions even for small to medium sized problem instances, in particular point-based and set-based minmax robust efficient solutions. An increase in the number of scenarios and objectives considered, as well as the size of the network, is associated with an increase in runtime. In case of the algorithms EL-SB and EL-PB, the values of the edge cost components also influence the runtime significantly. The experiments on instances with flimsily and highly robust efficiency show that it is preferable to use the class of repeated labeling algorithms for grid and many NetMaker instances, while extended labeling algorithms sometimes perform better for NetMaker networks with correlated scenarios.

https://doi.org/10.1002/net.21815

6 | CONCLUSION

In this article, we have investigated whether and how a generic label correcting algorithm for the multi-objective shortest path problem can be extended to find robust efficient solutions for the multi-objective uncertain shortest path problem. We have introduced algorithms to find robust efficient solutions for several popular concepts of robust efficiency, which can be classified into extended and repeated labeling algorithms. We compared their performance experimentally on several instances of grid networks and NetMaker networks and observed that the repeated labeling algorithms are often, but not always, faster than the extended labeling algorithms. We observed that in particular finding minmax robust efficient solutions is challenging even for small networks and few scenarios and objectives.

Therefore, investigating possible accelerations of the algorithms seems worthwhile, for example, by reducing the number of prediction matrices with the help of better upper bounds on the longest paths. More efficient ways to store and evaluate the information about the prediction matrices, than constructing one label per matrix, are also of interest.

There exists a great number of further concepts of robust efficiency, for example, lightly robust efficiency [24, 26, 39] and hull-based minmax robust efficiency [6]. The conditions for using the generic label correcting algorithm and the methods to extend it, as presented in this article, can also be useful when other concepts are considered.

The algorithm for the multi-objective problem that we have extended for the multi-objective uncertain case, is a generic algorithm with label selection. Our extended algorithms still include the label selection step. It would be of interest which label selection methods are best suited for the algorithms introduced in this article. In addition, the ideas presented to extend the label correcting algorithm with label selection might also be applicable to other labeling algorithms. Further research could also include possible extensions of other methods to solve the multi-objective or the robust shortest path problem.

In robust optimization, *PRO (Pareto robust optimal)* solutions are of interest (see [22] for single-objective, [26] for bi-objective problems and [7] for general multi-objective problems). PRO robust efficient solutions are solutions which are multi-scenario efficient and robust efficient w.r.t. some other concept at the same time. To find PRO robust efficient solutions, the approach given in [26] for bi-objective problems with uncertainty in only one objective can be extended to several uncertain objectives: First, one finds a complete set of multi-scenario efficient solutions, then these solutions are filtered to obtain the PRO robust efficient solutions. In comparison to the filtering procedure given in [26], filtering is much more time consuming for several uncertain objectives. Therefore, efficient filtering methods are of interest. In addition, pruning techniques would be useful, for example, as proposed in [32], where a multi-objective label correcting algorithm is used to find solutions of the single-objective minmax problem.

ACKNOWLEDGMENTS

Part of the implementations was done by Fabian Chlumsky-Harttmann. Lisa Thom was supported by DFG RTG 1703 “Resource Efficiency in Interorganizational Networks”.

ORCID

Andrea Raith http://orcid.org/0000-0002-0417-2972
Marie Schmidt http://orcid.org/0000-0001-9563-9955
Anita Schöbel http://orcid.org/0000-0002-9306-5529
Lisa Thom http://orcid.org/0000-0002-8589-6453

REFERENCES

[1] H. Aissi, C. Bazgan, and D. Vanderpooten, *Min–max and min–max regret versions of combinatorial optimization problems: A survey*, Eur. J. Oper. Res. **197** (2009), 427–438.

[2] R. Bellman, *On a routing problem*, Q. Appl. Math. **16** (1958), 87–90.

[3] H.P. Benson, *On a domination property for vector maximization with respect to cones*, J. Optim. Theory Appl. **39** (1983), 125–132.

[4] D. Bertsimas and M. Sim, *Robust discrete optimization and network flows*, Math. Program. **98** (2003), 49–71.

[5] G.R. Bitran, *Linear multiple objective problems with interval coefficients*, Manag. Sci. **26** (1980), 694–706.

[6] R. Bokrantz and A. Fredriksson, *Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization*, Eur. J. Oper. Res. **262** (2017), 682–692.

https://doi.org/10.1002/net.21815

[7] M. Botte and A. Schöbel, *Dominance for multi-objective robust optimization concepts*, Tech. report 2016-8, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2016.

[8] T.A. Brown and R.E. Strauch, *Dynamic programming in multiplicative lattices*, J. Math. Anal. Appl. **12** (1965), 364–370.

[9] J. Brumbaugh-Smith and D. Shier, *An empirical investigation of some bicriterion shortest path algorithms*, Eur. J. Oper. Res. **43** (1989), 216–224.

[10] A. Candia-Véjar, E. Álvarez-Miranda, and N. Maculan, *Minmax regret combinatorial optimization problems: An algorithmic perspective*, RAIRO Oper. Res. **45** (2011), 101–129.

[11] W.M. Carlyle and R.K. Wood, *Near-shortest and k-shortest simple paths*, Networks **46** (2005), 98–109.

[12] H.W. Corley and I.D. Moon, *Shortest paths in networks with vector weights*, J. Optim. Theory Appl. **46** (1985), 79–86.

[13] L.C. Dias and J.N. Clìmaco, *Shortest path problems with partial information: Models and algorithms for detecting dominance*, Eur. J. Oper. Res. **121** (2000), 16–31.

[14] E.W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math. **1** (1959), 269–271.

[15] T. Dokka and M. Goerigk, *Experimental evaluation of uncertainty sets for robust shortest path problem*, 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS2017) (G. D'Angelo and T. Dollevoet, eds.), OASIcs OpenAccess Ser. Inform., vol. 59, Schloss Dagstuhl. Leibniz-Zent. Inform., 2017, pp. 16:1–16:13.

[16] M. Ehrgott, *Multicriteria Optimization*, Springer, Berlin, 2005.

[17] M. Ehrgott, J. Ide, and A. Schöbel, *Minmax robustness for multi-objective optimization problems*, Eur. J. Oper. Res. **239** (2014), 17–31.

[18] G. Fadel et al., *Multi-criteria multi-scenario approaches in the design of vehicles*, 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[19] V. Gabrel and C. Murat, Robust shortest path problems, Annales du LAMSADE N 7 (2007), pp. 71–93.

[20] F. Guerriero and R. Musmanno, *Label correcting methods to solve multicriteria shortest path problems*, J. Optim. Theory Appl. **111** (2001), 589–613.

[21] M.I. Henig, *The principle of optimality in dynamic programming with returns in partially ordered sets*, Math. Oper. Res. **10** (1985), 462–470.

[22] D.A. Iancu and N. Trichakis, *Pareto efficiency in robust optimization*, Manag. Sci. **60** (2013), 130–147.

[23] J. Ide et al., *The relationship between multi-objective robustness concepts and set valued optimization*, Fixed Point Theory Appl. **2014** (2014), 83.

[24] J. Ide and A. Schöbel, *Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts*, OR Spectr. **38** (2016), 235–271.

[25] A. Kasperski and P. Zieliński, Robust discrete optimization under discrete and interval uncertainty: A survey, Robustness Analysis in Decision Aiding, Optimization, and Analytics (M. Doumpos, C. Zopounidis, and E. Grigoroudis, eds.), *International Series in Operations Research & Management Science*, Springer, Cham, 2016, pp. 113–143.

[26] K. Kuhn et al., *Bi-objective robust optimisation*, Eur. J. Oper. Res. **252** (2016), 418–431.

[27] D. Kuroiwa and G.M. Lee, *On robust multiobjective optimization*, Vietnam J. Math. **40** (2012), 305–317.

[28] E.Q.V. Martins, *On a multicriteria shortest path problem*, Eur. J. Oper. Res. **16** (1984), 236–245.

[29] L.G. Mitten, *Composition principles for synthesis of optimal multistage processes*, Oper. Res. **12** (1964), 610–619.

[30] R. Montemanni and L.M. Gambardella, *The robust shortest path problem with interval data via Benders decomposition*, 4OR **3** (2005), 315–328.

[31] T.L. Morin, *Monotonicity and the principle of optimality*, J. Math. Anal. Appl. **88** (1982), 665–674.

[32] I. Murthy and S. Her, *Solving min-max shortest-path problems on a network*, Naval Res. Logist. **39** (1992), 669–683.

[33] J.M. Paixão and J.L. Santos, Labeling methods for the general case of the multi-objective shortest path problem – a computational study, *Computational Intelligence and Decision Making* (A. Madureira, C. Reis, and V. Marques, eds.), Springer, Dordrecht, 2013, pp. 489–502.

[34] M. Pascoal and M. Resende, *The minmax regret robust shortest path problem in a finite multi-scenario model*, Appl. Math. Comput. **241** (2014), 88–111.

[35] P. Perny and O. Spanjaard, *A preference-based approach to spanning trees and shortest paths problems*, Eur. J. Oper. Res. **162** (2005), 584–601.

[36] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015.

https://doi.org/10.1002/net.21815

[37] A. Raith and M. Ehrgott, *A comparison of solution strategies for biobjective shortest path problems*, Comput. Oper. Res. **36** (2009), 1299–1331.

[38] A. Raith et al., *Multi-objective minmax robust combinatorial optimization with cardinality-constrained uncertainty*, Eur. J. Oper. Res. **267** (2018), 628–642.

[39] A. Schöbel, *Generalized light robustness and the trade-off between robustness and nominal quality*, Math. Methods Oper. Res. **80** (2014), 161–191.

[40] A.J.V. Skriver and K.A. Andersen, *A label correcting approach for solving bicriterion shortest-path problems*, Comput. Oper. Res. **27** (2000), 507–524.

[41] M. M. Wiecek and G. M. Dranichak, Robust multiobjective optimization for decision making under uncertainty and conflict, *Optimization Challenges in Complex, Networked and Risky Systems* (A. Gupta and A. Capponi, eds.), INFORMS Tutorials in Operations Research (J.C. Smith, ed.), INFORMS, Catonsville, MD, 2016, pp. 84–114.

[42] G. Yu and J. Yang, *On the robust shortest path problem*, Comput. Oper. Res. **25** (1998), 457–468.

How to cite this article: Raith A, Schmidt M, Schöbel A, Thom L. Extensions of labeling algorithms for multi-objective uncertain shortest path problems. *NETWORKS*. 2018;00:1–43. https://doi.org/10.1002/net.21815

https://doi.org/10.1002/net.21815

APPENDIX: RESULT TABLES

TABLE A1 Grid instances with two objectives ($k = 2$), correlated scenarios and $c = 10$, flimsily robust efficiency, 20 instances for each set of parameters

							EL-Flimsily			RL-Flimsily			sols		
	$\|V\|$	$\|E\|$	h	w	r	c	min	max	avg	min	max	avg	min	max	avg
1	103	380	10	10	2	10	0.00	0.00	0.00	0.00	0.00	0.00	4	22	12.65
3	103	380	10	10	4	10	0.01	0.01	0.01	0.00	0.00	0.00	12	34	19.60
5	103	380	10	10	6	10	0.01	0.04	0.02	0.00	0.01	0.01	8	43	27.75
7	103	380	10	10	8	10	0.02	0.05	0.03	0.01	0.01	0.01	14	56	35.80
9	203	760	10	20	2	10	0.02	0.04	0.03	0.01	0.02	0.01	22	42	30.95
11	203	760	10	20	4	10	0.07	0.15	0.11	0.01	0.05	0.03	33	74	53.45
13	203	760	10	20	6	10	0.07	0.45	0.28	0.02	0.06	0.04	43	140	87.05
15	203	760	10	20	8	10	0.20	0.77	0.43	0.03	0.10	0.06	59	173	94.70
17	303	1140	10	30	2	10	0.09	0.19	0.14	0.03	0.11	0.07	40	76	56.85
19	303	1140	10	30	4	10	0.30	0.96	0.59	0.08	0.26	0.14	61	145	100.90
21	303	1140	10	30	6	10	0.69	2.64	1.52	0.07	0.24	0.14	86	222	150.60
23	303	1140	10	30	8	10	1.74	5.16	3.01	0.12	0.28	0.18	143	260	199.60
25	403	1520	10	40	2	10	0.19	0.79	0.42	0.04	0.16	0.10	39	131	74.65
27	403	1520	10	40	4	10	0.82	3.10	1.76	0.12	0.35	0.19	95	213	146.00
29	403	1520	10	40	6	10	2.39	9.89	5.11	0.19	0.50	0.31	179	276	231.50
31	403	1520	10	40	8	10	4.85	21.50	12.98	0.22	0.72	0.47	167	451	318.55
33	203	780	20	10	2	10	0.01	0.01	0.01	0.00	0.01	0.00	5	23	14.35
35	203	780	20	10	4	10	0.01	0.03	0.02	0.01	0.01	0.01	10	41	24.85
37	203	780	20	10	6	10	0.03	0.06	0.04	0.01	0.02	0.01	16	51	35.10
39	203	780	20	10	8	10	0.05	0.10	0.07	0.01	0.03	0.02	15	69	41.80
41	403	1560	20	20	2	10	0.06	0.12	0.07	0.02	0.04	0.02	23	48	33.70
43	403	1560	20	20	4	10	0.14	0.53	0.26	0.03	0.09	0.05	44	110	67.25
45	403	1560	20	20	6	10	0.28	0.89	0.55	0.04	0.09	0.07	41	128	81.80
47	403	1560	20	20	8	10	0.56	1.93	1.08	0.07	0.16	0.10	54	137	103.65
49	603	2340	20	30	2	10	0.22	0.64	0.32	0.06	0.16	0.08	38	88	55.50
51	603	2340	20	30	4	10	0.96	1.84	1.42	0.12	0.21	0.17	71	132	106.05
53	603	2340	20	30	6	10	2.04	6.26	3.73	0.21	0.47	0.29	106	262	167.85
55	603	2340	20	30	8	10	4.23	15.76	8.00	0.25	0.55	0.39	144	310	217.25
57	803	3120	20	40	2	10	0.66	1.75	1.05	0.14	0.40	0.25	47	110	83.80
59	803	3120	20	40	4	10	2.74	6.81	4.68	0.34	0.74	0.51	112	252	168.95
61	803	3120	20	40	6	10	9.38	28.94	14.67	0.58	1.08	0.80	171	351	272.15
63	803	3120	20	40	8	10	17.70	44.44	26.97	0.75	1.61	1.02	231	449	322.10
65	303	1180	30	10	2	10	0.01	0.02	0.01	0.00	0.01	0.01	3	23	15.00
67	303	1180	30	10	4	10	0.02	0.05	0.03	0.01	0.01	0.01	10	42	24.25
69	303	1180	30	10	6	10	0.05	0.09	0.07	0.01	0.02	0.02	22	44	33.65
71	303	1180	30	10	8	10	0.07	0.13	0.10	0.02	0.03	0.02	26	51	39.55
73	603	2360	30	20	2	10	0.09	0.19	0.12	0.03	0.05	0.04	18	51	35.75
75	603	2360	30	20	4	10	0.25	0.52	0.41	0.05	0.09	0.07	43	97	61.10
77	603	2360	30	20	6	10	0.60	1.76	1.00	0.09	0.17	0.12	36	135	88.45
79	603	2360	30	20	8	10	1.18	3.42	1.80	0.12	0.22	0.16	45	167	116.10
81	903	3540	30	30	2	10	0.34	0.75	0.51	0.09	0.19	0.13	39	82	54.80
83	903	3540	30	30	4	10	1.59	2.70	2.18	0.20	0.34	0.28	84	145	111.35
85	903	3540	30	30	6	10	4.13	8.61	5.50	0.33	0.48	0.41	94	218	165.40
87	903	3540	30	30	8	10	7.26	15.39	10.80	0.47	0.74	0.59	133	264	215.70
89	1203	4720	30	40	2	10	1.04	2.28	1.55	0.23	0.46	0.33	56	110	81.65
91	1203	4720	30	40	4	10	5.83	11.03	8.60	0.57	1.11	0.84	140	248	188.70
93	1203	4720	30	40	6	10	15.52	29.63	22.92	1.04	1.67	1.28	187	360	271.10
95	1203	4720	30	40	8	10	31.81	63.96	48.12	1.15	2.72	1.77	255	458	349.45
97	403	1580	40	10	2	10	0.01	0.02	0.02	0.01	0.01	0.01	9	26	15.85
99	403	1580	40	10	4	10	0.04	0.06	0.05	0.01	0.02	0.02	11	40	26.10
101	403	1580	40	10	6	10	0.07	0.12	0.09	0.02	0.04	0.03	17	46	31.55
103	403	1580	40	10	8	10	0.10	0.18	0.14	0.03	0.04	0.03	31	67	43.60
105	803	3160	40	20	2	10	0.11	0.23	0.15	0.03	0.07	0.05	17	58	36.70
107	803	3160	40	20	4	10	0.48	0.73	0.57	0.09	0.12	0.10	41	93	71.15
109	803	3160	40	20	6	10	1.03	1.70	1.35	0.14	0.20	0.17	71	128	91.65
111	803	3160	40	20	8	10	1.76	2.98	2.30	0.18	0.27	0.21	67	149	111.30
113	1203	4740	40	30	2	10	0.51	1.21	0.74	0.12	0.31	0.18	35	90	61.70
115	1203	4740	40	30	4	10	1.65	3.63	2.83	0.24	0.44	0.36	68	147	111.10
117	1203	4740	40	30	6	10	5.48	10.65	7.69	0.46	0.82	0.61	101	194	159.20
119	1203	4740	40	30	8	10	10.39	19.77	15.31	0.59	0.97	0.81	106	286	217.55
121	1603	6320	40	40	2	10	1.32	2.91	2.25	0.30	0.62	0.51	55	116	86.90
123	1603	6320	40	40	4	10	7.94	17.02	10.63	0.86	1.69	1.10	121	263	179.45
125	1603	6320	40	40	6	10	23.53	39.65	30.27	1.40	2.08	1.73	162	339	261.65
127	1603	6320	40	40	8	10	38.21	88.07	58.58	1.52	3.24	2.20	251	526	337.95

https://doi.org/10.1002/net.21815

TABLE A2 Grid instances with two objectives ($k = 2$), correlated scenarios and $c = 100$, flimsily robust efficiency, 20 instances for each set of parameters

							EL-Flimsily			RL-Flimsily			sols		
	$\|V\|$	$\|E\|$	*h*	*w*	*r*	*c*	min	max	avg	min	max	avg	min	max	avg
129	103	380	10	10	2	100	0.00	0.01	0.00	0.00	0.00	0.00	4	24	11.80
131	103	380	10	10	4	100	0.00	0.01	0.01	0.00	0.01	0.01	4	21	14.35
133	103	380	10	10	6	100	0.01	0.02	0.01	0.01	0.02	0.01	7	26	16.35
135	103	380	10	10	8	100	0.01	0.01	0.01	0.01	0.02	0.01	6	28	17.25
137	203	760	10	20	2	100	0.02	0.06	0.03	0.01	0.05	0.02	11	74	30.45
139	203	760	10	20	4	100	0.03	0.10	0.06	0.02	0.08	0.04	20	83	36.55
141	203	760	10	20	6	100	0.04	0.14	0.09	0.03	0.11	0.06	10	76	35.20
143	203	760	10	20	8	100	0.05	0.29	0.12	0.05	0.21	0.09	21	61	40.85
145	303	1140	10	30	2	100	0.08	0.24	0.14	0.04	0.13	0.08	27	71	49.20
147	303	1140	10	30	4	100	0.11	0.40	0.25	0.07	0.36	0.17	35	92	62.95
149	303	1140	10	30	6	100	0.18	1.05	0.44	0.10	0.57	0.28	41	118	78.45
151	303	1140	10	30	8	100	0.11	1.04	0.49	0.05	0.59	0.32	16	134	71.20
153	403	1520	10	40	2	100	0.15	1.05	0.44	0.10	0.59	0.27	49	154	88.35
155	403	1520	10	40	4	100	0.41	5.05	1.09	0.24	1.97	0.61	62	258	118.85
157	403	1520	10	40	6	100	0.56	3.18	1.68	0.36	1.42	0.91	63	206	122.20
159	403	1520	10	40	8	100	0.91	4.36	2.16	0.51	1.70	0.98	71	264	148.25
161	203	780	20	10	2	100	0.01	0.01	0.01	0.00	0.01	0.01	8	26	15.75
163	203	780	20	10	4	100	0.01	0.02	0.01	0.01	0.02	0.01	8	28	15.45
165	203	780	20	10	6	100	0.02	0.04	0.02	0.01	0.03	0.02	13	39	21.65
167	203	780	20	10	8	100	0.01	0.05	0.02	0.02	0.05	0.03	9	24	15.60
169	403	1560	20	20	2	100	0.04	0.22	0.08	0.03	0.11	0.05	20	63	32.10
171	403	1560	20	20	4	100	0.07	0.26	0.15	0.05	0.17	0.10	23	71	43.25
173	403	1560	20	20	6	100	0.11	0.29	0.19	0.08	0.25	0.14	18	73	38.30
175	403	1560	20	20	8	100	0.17	0.51	0.27	0.10	0.45	0.23	34	72	48.95
177	603	2340	20	30	2	100	0.19	0.78	0.37	0.11	0.45	0.23	34	96	60.65
179	603	2340	20	30	4	100	0.33	1.18	0.65	0.26	0.80	0.44	41	123	79.60
181	603	2340	20	30	6	100	0.59	2.53	1.17	0.43	1.15	0.69	50	132	87.40
183	603	2340	20	30	8	100	0.69	2.09	1.17	0.41	1.35	0.75	46	138	87.20
185	803	3120	20	40	2	100	0.76	2.03	1.19	0.40	1.23	0.65	60	129	96.95
187	803	3120	20	40	4	100	0.98	3.79	2.22	0.44	1.87	1.17	52	142	105.70
189	803	3120	20	40	6	100	1.63	9.19	4.05	1.17	5.95	2.19	63	263	135.80
191	803	3120	20	40	8	100	3.55	9.19	5.55	1.45	5.03	2.67	89	251	148.85
193	303	1180	30	10	2	100	0.01	0.02	0.01	0.01	0.02	0.01	6	23	13.85
195	303	1180	30	10	4	100	0.02	0.03	0.02	0.02	0.03	0.02	8	31	17.70
197	303	1180	30	10	6	100	0.02	0.04	0.03	0.02	0.04	0.03	11	30	18.10
199	303	1180	30	10	8	100	0.03	0.05	0.04	0.03	0.05	0.04	7	31	17.65
201	603	2360	30	20	2	100	0.07	0.17	0.11	0.04	0.11	0.07	10	56	30.20
203	603	2360	30	20	4	100	0.11	0.36	0.21	0.09	0.23	0.15	18	51	38.15
205	603	2360	30	20	6	100	0.19	0.46	0.33	0.14	0.36	0.24	25	95	50.15
207	603	2360	30	20	8	100	0.27	0.60	0.40	0.18	0.51	0.30	24	72	47.60
209	903	3540	30	30	2	100	0.37	0.74	0.55	0.18	0.47	0.30	38	84	59.05
211	903	3540	30	30	4	100	0.74	2.17	1.15	0.43	1.23	0.69	36	115	74.55
213	903	3540	30	30	6	100	1.20	2.49	1.76	0.67	1.51	1.04	36	136	92.20
215	903	3540	30	30	8	100	1.27	3.63	2.11	0.80	2.86	1.38	49	156	89.55
217	1203	4720	30	40	2	100	0.75	2.80	1.90	0.37	1.88	1.02	40	168	89.40
219	1203	4720	30	40	4	100	2.62	6.06	4.24	1.24	3.62	2.17	84	168	125.35
221	1203	4720	30	40	6	100	4.11	9.75	6.48	2.28	4.53	3.15	76	217	145.15
223	1203	4720	30	40	8	100	5.32	10.74	7.90	2.60	6.75	4.28	83	279	138.30
225	403	1580	40	10	2	100	0.01	0.03	0.02	0.01	0.02	0.01	8	24	14.60
227	403	1580	40	10	4	100	0.02	0.04	0.03	0.02	0.04	0.02	8	30	17.75
229	403	1580	40	10	6	100	0.03	0.06	0.04	0.03	0.07	0.04	9	32	15.90
231	403	1580	40	10	8	100	0.04	0.08	0.05	0.04	0.10	0.06	13	30	20.65
233	803	3160	40	20	2	100	0.11	0.25	0.16	0.07	0.16	0.10	22	63	33.65
235	803	3160	40	20	4	100	0.19	0.42	0.28	0.15	0.37	0.22	18	65	43.40
237	803	3160	40	20	6	100	0.37	0.71	0.47	0.25	0.48	0.36	31	99	51.95
239	803	3160	40	20	8	100	0.34	0.64	0.51	0.26	0.47	0.40	18	72	45.30
241	1203	4740	40	30	2	100	0.41	1.38	0.77	0.27	0.58	0.43	40	85	56.65
243	1203	4740	40	30	4	100	0.90	1.65	1.28	0.59	1.17	0.86	27	115	74.60
245	1203	4740	40	30	6	100	1.53	3.90	2.33	0.92	1.94	1.40	59	136	89.50
247	1203	4740	40	30	8	100	1.85	5.02	2.97	1.15	3.26	1.88	47	165	95.90
249	1603	6320	40	40	2	100	1.84	4.62	2.59	0.97	2.38	1.46	68	195	105.50
251	1603	6320	40	40	4	100	4.03	10.19	5.72	2.06	5.87	3.36	81	270	131.80
253	1603	6320	40	40	6	100	4.26	13.71	8.52	2.68	8.51	4.75	82	216	148.90
255	1603	6320	40	40	8	100	8.58	20.71	12.98	4.10	9.75	6.46	75	260	173.40

https://doi.org/10.1002/net.21815

TABLE A3 Grid instances with two objectives ($k = 2$), correlated scenarios and $c = 10$, highly robust efficiency, 20 instances for each set of parameters

							EL-Highly			RL-Highly			sols		
	$\|V\|$	$\|E\|$	h	w	r	c	min	max	avg	min	max	avg	min	max	avg
1	103	380	10	10	2	10	0.00	0.01	0.00	0.00	0.00	0.00	1	9	3.60
3	103	380	10	10	4	10	0.01	0.01	0.01	0.00	0.00	0.00	0	3	1.10
5	103	380	10	10	6	10	0.01	0.04	0.02	0.00	0.01	0.01	0	3	0.80
7	103	380	10	10	8	10	0.02	0.05	0.03	0.01	0.01	0.01	0	3	0.40
9	203	760	10	20	2	10	0.02	0.04	0.03	0.01	0.02	0.01	0	6	2.90
11	203	760	10	20	4	10	0.07	0.15	0.11	0.01	0.04	0.02	0	3	0.80
13	203	760	10	20	6	10	0.07	0.46	0.28	0.02	0.06	0.04	0	1	0.15
15	203	760	10	20	8	10	0.20	0.79	0.43	0.03	0.07	0.04	0	1	0.10
17	303	1140	10	30	2	10	0.09	0.19	0.14	0.02	0.06	0.04	0	9	4.20
19	303	1140	10	30	4	10	0.30	0.98	0.60	0.05	0.12	0.08	0	1	0.15
21	303	1140	10	30	6	10	0.71	2.64	1.55	0.06	0.23	0.13	0	0	0.00
23	303	1140	10	30	8	10	1.81	5.23	3.11	0.11	0.25	0.17	0	0	0.00
25	403	1520	10	40	2	10	0.19	0.74	0.42	0.04	0.16	0.10	0	7	2.70
27	403	1520	10	40	4	10	0.82	3.17	1.78	0.11	0.35	0.19	0	1	0.20
29	403	1520	10	40	6	10	2.48	10.35	5.29	0.19	0.51	0.31	0	0	0.00
31	403	1520	10	40	8	10	4.93	22.29	13.44	0.22	0.70	0.46	0	0	0.00
33	203	780	20	10	2	10	0.01	0.01	0.01	0.00	0.00	0.00	1	8	3.60
35	203	780	20	10	4	10	0.01	0.03	0.02	0.01	0.01	0.01	0	4	0.90
37	203	780	20	10	6	10	0.03	0.06	0.04	0.01	0.02	0.01	0	1	0.25
39	203	780	20	10	8	10	0.05	0.10	0.07	0.01	0.02	0.02	0	1	0.10
41	403	1560	20	20	2	10	0.06	0.12	0.07	0.02	0.04	0.02	1	9	3.30
43	403	1560	20	20	4	10	0.14	0.52	0.27	0.03	0.09	0.05	0	2	0.30
45	403	1560	20	20	6	10	0.29	0.91	0.57	0.04	0.09	0.07	0	1	0.10
47	403	1560	20	20	8	10	0.58	1.97	1.10	0.06	0.16	0.10	0	1	0.10
49	603	2340	20	30	2	10	0.22	0.64	0.33	0.06	0.16	0.08	0	7	2.95
51	603	2340	20	30	4	10	0.97	1.89	1.44	0.12	0.21	0.17	0	2	0.35
53	603	2340	20	30	6	10	2.10	6.39	3.84	0.21	0.47	0.29	0	1	0.05
55	603	2340	20	30	8	10	4.39	16.13	8.50	0.25	0.55	0.38	0	0	0.00
57	803	3120	20	40	2	10	0.65	1.77	1.06	0.14	0.39	0.25	0	12	3.65
59	803	3120	20	40	4	10	2.81	6.82	4.78	0.33	0.73	0.51	0	3	0.20
61	803	3120	20	40	6	10	9.61	29.69	14.98	0.56	1.09	0.80	0	0	0.00
63	803	3120	20	40	8	10	18.01	45.68	27.57	0.73	1.60	1.02	0	0	0.00
65	303	1180	30	10	2	10	0.01	0.02	0.01	0.00	0.01	0.01	0	5	2.95
67	303	1180	30	10	4	10	0.02	0.05	0.03	0.01	0.01	0.01	0	3	1.10
69	303	1180	30	10	6	10	0.05	0.09	0.07	0.01	0.02	0.02	0	2	0.50
71	303	1180	30	10	8	10	0.07	0.13	0.10	0.02	0.03	0.02	0	1	0.20
73	603	2360	30	20	2	10	0.09	0.19	0.12	0.03	0.05	0.04	0	9	4.20
75	603	2360	30	20	4	10	0.26	0.54	0.42	0.05	0.09	0.07	0	1	0.15
77	603	2360	30	20	6	10	0.62	1.94	1.03	0.08	0.17	0.12	0	2	0.25
79	603	2360	30	20	8	10	1.23	3.44	1.86	0.12	0.22	0.16	0	2	0.15
81	903	3540	30	30	2	10	0.34	0.76	0.52	0.08	0.19	0.13	0	7	2.65
83	903	3540	30	30	4	10	1.69	2.77	2.21	0.20	0.33	0.28	0	1	0.15
85	903	3540	30	30	6	10	4.20	8.71	5.54	0.33	0.48	0.41	0	1	0.05
87	903	3540	30	30	8	10	7.24	15.50	10.88	0.47	0.74	0.58	0	0	0.00
89	1203	4720	30	40	2	10	1.05	2.23	1.55	0.23	0.45	0.33	0	9	2.60
91	1203	4720	30	40	4	10	5.59	11.24	8.70	0.57	1.12	0.86	0	1	0.05
93	1203	4720	30	40	6	10	15.87	29.90	23.33	1.01	1.66	1.27	0	0	0.00
95	1203	4720	30	40	8	10	32.53	65.46	49.26	1.15	2.82	1.75	0	0	0.00
97	403	1580	40	10	2	10	0.01	0.02	0.02	0.01	0.01	0.01	2	6	3.95
99	403	1580	40	10	4	10	0.04	0.06	0.05	0.01	0.02	0.02	0	3	0.95
101	403	1580	40	10	6	10	0.07	0.13	0.09	0.02	0.03	0.02	0	1	0.60
103	403	1580	40	10	8	10	0.10	0.18	0.15	0.02	0.04	0.03	0	2	0.25
105	803	3160	40	20	2	10	0.11	0.23	0.16	0.03	0.07	0.05	0	7	3.15
107	803	3160	40	20	4	10	0.48	0.73	0.58	0.09	0.12	0.10	0	3	0.25
109	803	3160	40	20	6	10	0.95	1.70	1.37	0.14	0.21	0.17	0	2	0.15
111	803	3160	40	20	8	10	1.78	2.96	2.34	0.18	0.28	0.21	0	0	0.00
113	1203	4740	40	30	2	10	0.50	1.22	0.73	0.12	0.31	0.18	0	9	3.40
115	1203	4740	40	30	4	10	1.68	3.74	2.89	0.24	0.45	0.36	0	3	0.35
117	1203	4740	40	30	6	10	5.68	11.10	7.88	0.46	0.84	0.60	0	1	0.05
119	1203	4740	40	30	8	10	10.19	20.34	15.68	0.61	0.99	0.80	0	0	0.00
121	1603	6320	40	40	2	10	1.33	2.86	2.27	0.29	0.60	0.51	0	7	3.25
123	1603	6320	40	40	4	10	8.09	16.87	10.71	0.86	1.66	1.09	0	1	0.10
125	1603	6320	40	40	6	10	23.93	40.16	30.81	1.37	2.03	1.71	0	1	0.05
127	1603	6320	40	40	8	10	38.91	90.21	59.36	1.51	3.13	2.23	0	0	0.00

https://doi.org/10.1002/net.21815

TABLE A4 Grid instances with two objectives ($k = 2$), correlated scenarios and $c = 100$, highly robust efficiency, 20 instances for each set of parameters

							EL-Highly			RL-Highly			sols		
	$\|V\|$	$\|E\|$	h	w	r	c	min	max	avg	min	max	avg	min	max	avg
129	103	380	10	10	2	100	0.00	0.01	0.00	0.00	0.00	0.00	4	18	9.75
131	103	380	10	10	4	100	0.00	0.01	0.01	0.00	0.01	0.01	4	11	8.45
133	103	380	10	10	6	100	0.01	0.02	0.01	0.01	0.02	0.01	5	13	9.50
135	103	380	10	10	8	100	0.01	0.02	0.01	0.01	0.02	0.01	4	12	8.20
137	203	760	10	20	2	100	0.02	0.07	0.03	0.01	0.05	0.02	7	39	20.90
139	203	760	10	20	4	100	0.03	0.11	0.06	0.02	0.08	0.04	10	32	17.55
141	203	760	10	20	6	100	0.04	0.14	0.09	0.03	0.11	0.07	5	22	13.50
143	203	760	10	20	8	100	0.05	0.29	0.12	0.05	0.21	0.09	7	20	14.55
145	303	1140	10	30	2	100	0.09	0.23	0.14	0.04	0.13	0.09	19	55	34.40
147	303	1140	10	30	4	100	0.11	0.39	0.24	0.07	0.33	0.17	15	33	25.65
149	303	1140	10	30	6	100	0.18	1.01	0.44	0.11	0.59	0.29	13	32	23.70
151	303	1140	10	30	8	100	0.12	1.02	0.49	0.05	0.58	0.33	8	27	18.70
153	403	1520	10	40	2	100	0.15	1.04	0.43	0.10	0.58	0.26	32	94	49.65
155	403	1520	10	40	4	100	0.41	5.01	1.07	0.20	1.90	0.57	24	61	34.70
157	403	1520	10	40	6	100	0.56	3.22	1.67	0.27	1.41	0.85	4	40	28.05
159	403	1520	10	40	8	100	0.91	4.40	2.18	0.53	1.84	1.02	17	36	24.95
161	203	780	20	10	2	100	0.01	0.02	0.01	0.00	0.01	0.01	7	21	12.35
163	203	780	20	10	4	100	0.01	0.02	0.01	0.01	0.02	0.01	5	14	9.45
165	203	780	20	10	6	100	0.01	0.04	0.02	0.01	0.03	0.02	6	15	9.80
167	203	780	20	10	8	100	0.01	0.04	0.02	0.01	0.04	0.02	5	13	8.10
169	403	1560	20	20	2	100	0.04	0.20	0.08	0.03	0.11	0.05	13	45	23.00
171	403	1560	20	20	4	100	0.07	0.25	0.14	0.05	0.18	0.10	13	28	19.35
173	403	1560	20	20	6	100	0.12	0.30	0.19	0.08	0.22	0.15	3	22	13.65
175	403	1560	20	20	8	100	0.17	0.51	0.27	0.10	0.37	0.22	9	25	16.70
177	603	2340	20	30	2	100	0.19	0.87	0.38	0.11	0.54	0.25	23	52	37.75
179	603	2340	20	30	4	100	0.33	1.18	0.65	0.25	0.80	0.43	19	48	29.60
181	603	2340	20	30	6	100	0.57	2.59	1.16	0.43	1.07	0.68	8	32	24.30
183	603	2340	20	30	8	100	0.66	2.05	1.17	0.41	1.29	0.73	10	38	22.05
185	803	3120	20	40	2	100	0.77	1.99	1.19	0.38	1.13	0.64	28	73	54.75
187	803	3120	20	40	4	100	0.98	3.77	2.21	0.43	1.56	1.13	22	49	34.10
189	803	3120	20	40	6	100	1.65	9.11	4.07	1.00	4.65	2.03	19	47	32.80
191	803	3120	20	40	8	100	3.48	9.35	5.52	1.42	4.84	2.63	13	49	25.90
193	303	1180	30	10	2	100	0.01	0.02	0.01	0.01	0.02	0.01	6	19	10.70
195	303	1180	30	10	4	100	0.02	0.03	0.02	0.01	0.03	0.02	6	15	10.20
197	303	1180	30	10	6	100	0.02	0.05	0.03	0.02	0.04	0.03	5	14	8.95
199	303	1180	30	10	8	100	0.02	0.05	0.04	0.03	0.05	0.04	3	12	8.20
201	603	2360	30	20	2	100	0.07	0.15	0.11	0.04	0.11	0.07	9	32	22.00
203	603	2360	30	20	4	100	0.11	0.36	0.20	0.09	0.24	0.16	13	24	19.35
205	603	2360	30	20	6	100	0.19	0.43	0.32	0.14	0.39	0.24	13	25	18.25
207	603	2360	30	20	8	100	0.27	0.58	0.40	0.20	0.56	0.32	4	28	14.35
209	903	3540	30	30	2	100	0.36	0.74	0.54	0.18	0.47	0.30	25	51	37.30
211	903	3540	30	30	4	100	0.71	2.19	1.15	0.43	1.45	0.73	19	35	27.60
213	903	3540	30	30	6	100	1.20	2.51	1.75	0.67	1.61	1.08	16	33	22.90
215	903	3540	30	30	8	100	1.26	3.73	2.13	0.81	1.97	1.26	9	26	18.80
217	1203	4720	30	40	2	100	0.75	2.78	1.91	0.38	1.83	1.03	30	92	52.50
219	1203	4720	30	40	4	100	2.57	6.36	4.22	1.24	4.01	2.20	26	48	36.10
221	1203	4720	30	40	6	100	4.25	9.84	6.53	2.30	5.45	3.63	19	42	28.50
223	1203	4720	30	40	8	100	5.28	10.82	7.92	2.54	7.54	4.36	16	41	28.20
225	403	1580	40	10	2	100	0.01	0.02	0.02	0.01	0.02	0.01	7	19	11.35
227	403	1580	40	10	4	100	0.02	0.05	0.03	0.02	0.04	0.03	5	16	10.20
229	403	1580	40	10	6	100	0.03	0.07	0.04	0.03	0.06	0.04	3	15	8.20
231	403	1580	40	10	8	100	0.04	0.08	0.05	0.04	0.07	0.06	6	13	8.40
233	803	3160	40	20	2	100	0.13	0.25	0.16	0.08	0.14	0.11	15	43	23.40
235	803	3160	40	20	4	100	0.19	0.42	0.29	0.15	0.36	0.23	7	29	19.50
237	803	3160	40	20	6	100	0.37	0.75	0.48	0.29	0.46	0.37	13	25	17.55
239	803	3160	40	20	8	100	0.35	0.62	0.51	0.25	0.53	0.42	8	17	13.60
241	1203	4740	40	30	2	100	0.42	1.41	0.78	0.24	0.78	0.44	26	49	37.30
243	1203	4740	40	30	4	100	0.93	1.63	1.29	0.59	1.11	0.79	15	46	27.65
245	1203	4740	40	30	6	100	1.56	3.82	2.34	0.91	2.15	1.36	15	32	23.50
247	1203	4740	40	30	8	100	1.85	5.22	3.01	1.12	3.26	1.74	15	29	20.10
249	1603	6320	40	40	2	100	1.85	4.82	2.63	0.96	2.31	1.39	36	114	59.00
251	1603	6320	40	40	4	100	4.08	10.29	5.79	2.06	4.72	2.89	7	58	36.35
253	1603	6320	40	40	6	100	4.23	14.13	8.56	2.18	5.67	4.20	19	41	30.90
255	1603	6320	40	40	8	100	8.91	21.57	13.32	3.91	9.26	6.08	11	40	24.15

https://doi.org/10.1002/net.21815

TABLE A5 Grid instances with three objectives ($k = 3$), correlated scenarios and $c = 10$

	|V|	|E|	h	w	r	c	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
1025	103	380	10	10	2	10	0.04	0.03	72	0.02	0.01	21
1027	103	380	10	10	4	10	0.09	0.04	96	0.07	0.02	7
1029	103	380	10	10	6	10	0.11	0.03	69	0.10	0.03	2
1031	103	380	10	10	8	10	0.15	0.04	105	0.15	0.04	2
1033	203	760	10	20	2	10	0.27	0.08	155	0.26	0.09	26
1035	203	760	10	20	4	10	1.25	0.19	240	1.20	0.20	5
1037	203	760	10	20	6	10	3.19	0.38	283	3.07	0.38	0
1039	203	760	10	20	8	10	6.55	0.56	510	6.14	0.56	2
1041	303	1140	10	30	2	10	5.56	1.64	490	5.57	1.64	42
1043	303	1140	10	30	4	10	16.03	2.28	996	16.77	2.26	1
1045	303	1140	10	30	6	10	60.32	2.96	1078	50.14	2.98	3
1047	303	1140	10	30	8	10	42.26	2.45	808	43.13	2.51	1
1049	403	1520	10	40	2	10	21.81	5.15	634	22.33	5.30	41
1051	403	1520	10	40	4	10	209.11	15.87	1723	202.09	16.00	9
1053	403	1520	10	40	6	10	416.58	14.04	2081	404.02	12.99	0
1055	403	1520	10	40	8	10	1018.44	29.61	4179	1005.64	24.91	1
1057	203	780	20	10	2	10	0.02	0.01	35	0.02	0.01	6
1059	203	780	20	10	4	10	0.13	0.04	63	0.13	0.04	3
1061	203	780	20	10	6	10	0.20	0.06	90	0.23	0.06	3
1063	203	780	20	10	8	10	0.25	0.07	135	0.24	0.06	2
1065	403	1560	20	20	2	10	1.17	0.58	209	1.19	0.47	40
1067	403	1560	20	20	4	10	3.51	0.74	455	3.41	0.62	7
1069	403	1560	20	20	6	10	12.03	1.33	708	12.25	1.25	0
1071	403	1560	20	20	8	10	15.46	1.13	544	15.60	1.07	1
1073	603	2340	20	30	2	10	8.78	2.16	355	8.33	2.15	7
1075	603	2340	20	30	4	10	68.36	5.17	597	51.39	5.22	5
1077	603	2340	20	30	6	10	137.32	8.45	1116	126.29	7.76	3
1079	603	2340	20	30	8	10	290.04	11.84	1173	278.94	11.09	1
1081	803	3120	20	40	2	10	65.09	22.59	927	53.52	17.63	60
1083	803	3120	20	40	4	10	710.31	71.53	1740	627.12	62.86	2
1085	803	3120	20	40	6	10	936.21	50.42	1615	822.91	46.90	2
1087	803	3120	20	40	8	10	2943.12	114.73	4074	2854.50	94.95	0
1089	303	1180	30	10	2	10	0.10	0.04	114	0.10	0.05	34
1091	303	1180	30	10	4	10	0.22	0.07	145	0.23	0.08	7
1093	303	1180	30	10	6	10	0.41	0.10	112	0.42	0.11	3
1095	303	1180	30	10	8	10	0.76	0.14	222	0.76	0.15	2
1097	603	2360	30	20	2	10	1.66	0.56	230	2.11	0.56	49
1099	603	2360	30	20	4	10	8.85	1.51	422	8.66	1.54	11
1101	603	2360	30	20	6	10	19.84	1.96	499	17.40	1.98	5
1103	603	2360	30	20	8	10	32.78	2.14	570	29.51	2.17	1
1105	903	3540	30	30	2	10	14.47	4.77	443	15.74	4.80	41
1107	903	3540	30	30	4	10	69.30	10.08	986	68.99	8.43	3
1109	903	3540	30	30	6	10	481.32	30.90	1513	432.96	24.93	2
1111	903	3540	30	30	8	10	742.98	39.42	2263	778.46	31.54	0
1113	1203	4720	30	40	2	10	110.40	41.12	1582	113.89	36.97	45
1115	1203	4720	30	40	4	10	715.42	98.19	2273	751.41	87.72	1
1117	1203	4720	30	40	6	10	2606.18	163.44	3579	2782.03	144.30	0
1119	1203	4720	30	40	8	10	> 3600.00	194.80	3551	> 3600.00	165.90	1
1121	403	1580	40	10	2	10	0.08	0.03	54	0.08	0.04	12
1123	403	1580	40	10	4	10	0.23	0.07	60	0.27	0.08	4
1125	403	1580	40	10	6	10	0.64	0.15	113	0.62	0.16	4
1127	403	1580	40	10	8	10	0.97	0.19	175	0.99	0.19	2
1129	803	3160	40	20	2	10	1.47	0.68	179	1.54	0.53	27
1131	803	3160	40	20	4	10	5.48	1.22	160	5.63	1.00	7
1133	803	3160	40	20	6	10	24.71	2.96	524	28.47	2.58	1
1135	803	3160	40	20	8	10	46.75	4.61	715	53.54	3.77	2
1137	1203	4740	40	30	2	10	22.17	6.77	437	23.59	6.67	44
1139	1203	4740	40	30	4	10	97.76	17.14	1033	106.29	15.78	5
1141	1203	4740	40	30	6	10	499.83	38.67	1973	535.59	34.74	0
1143	1203	4740	40	30	8	10	522.77	32.17	1823	566.18	25.82	2
1145	1603	6320	40	40	2	10	198.73	73.84	1038	217.47	71.63	84
1147	1603	6320	40	40	4	10	837.58	114.30	1661	898.48	99.70	3
1149	1603	6320	40	40	6	10	2962.03	225.56	2900	3317.24	199.17	2
1151	1603	6320	40	40	8	10	> 3600.00	189.83	3069	> 3600.00	175.23	1

https://doi.org/10.1002/net.21815

TABLE A6 Grid instances with three objectives ($k = 3$), correlated scenarios and $c = 100$

	$\lvert V\rvert$	$\lvert E\rvert$	*h*	*w*	*r*	*c*	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
1153	103	380	10	10	2	100	0.01	0.01	25	0.01	0.01	23
1155	103	380	10	10	4	100	0.03	0.04	58	0.03	0.05	39
1157	103	380	10	10	6	100	0.06	0.06	88	0.07	0.07	52
1159	103	380	10	10	8	100	0.02	0.02	23	0.02	0.02	13
1161	203	760	10	20	2	100	0.64	0.49	250	0.64	0.50	159
1163	203	760	10	20	4	100	1.23	1.41	371	1.28	1.40	215
1165	203	760	10	20	6	100	1.29	1.09	412	1.31	1.09	150
1167	203	760	10	20	8	100	0.74	0.60	114	0.78	0.63	42
1169	303	1140	10	30	2	100	5.71	5.20	478	6.06	5.09	341
1171	303	1140	10	30	4	100	60.49	36.72	1653	70.41	36.67	597
1173	303	1140	10	30	6	100	22.42	15.80	736	23.18	15.84	279
1175	303	1140	10	30	8	100	24.33	17.33	997	27.75	17.39	244
1177	403	1520	10	40	2	100	30.76	30.02	1211	35.43	30.37	733
1179	403	1520	10	40	4	100	171.00	100.59	1552	199.69	100.81	497
1181	403	1520	10	40	6	100	399.05	213.91	2050	425.82	211.03	552
1183	403	1520	10	40	8	100	197.30	112.93	1202	193.04	102.14	256
1185	203	780	20	10	2	100	0.07	0.06	113	0.07	0.10	92
1187	203	780	20	10	4	100	0.08	0.10	81	0.08	0.10	57
1189	203	780	20	10	6	100	0.10	0.12	74	0.10	0.11	44
1191	203	780	20	10	8	100	0.12	0.17	30	0.12	0.13	17
1193	403	1560	20	20	2	100	0.98	0.95	258	0.97	0.75	187
1195	403	1560	20	20	4	100	1.04	1.20	204	1.01	0.98	103
1197	403	1560	20	20	6	100	1.66	2.12	320	1.65	1.52	126
1199	403	1560	20	20	8	100	4.16	3.93	279	4.12	3.04	114
1201	603	2340	20	30	2	100	8.45	7.01	359	7.43	6.43	255
1203	603	2340	20	30	4	100	35.34	25.79	613	28.75	22.23	268
1205	603	2340	20	30	6	100	234.84	153.17	1558	196.22	136.94	492
1207	603	2340	20	30	8	100	91.63	84.96	932	82.78	64.63	237
1209	803	3120	20	40	2	100	158.33	148.63	1530	150.82	138.40	1058
1211	803	3120	20	40	4	100	302.19	203.66	1543	281.19	200.48	547
1213	803	3120	20	40	6	100	942.37	507.00	1939	838.62	515.52	507
1215	803	3120	20	40	8	100	402.88	261.81	1982	383.54	231.58	415
1217	303	1180	30	10	2	100	0.08	0.06	90	0.07	0.06	75
1219	303	1180	30	10	4	100	0.16	0.17	83	0.15	0.20	41
1221	303	1180	30	10	6	100	0.16	0.18	118	0.15	0.18	56
1223	303	1180	30	10	8	100	0.17	0.21	73	0.17	0.18	38
1225	603	2360	30	20	2	100	2.29	1.67	264	2.26	1.67	192
1227	603	2360	30	20	4	100	3.03	2.89	255	2.98	2.90	111
1229	603	2360	30	20	6	100	4.69	3.84	313	4.58	3.81	117
1231	603	2360	30	20	8	100	10.11	8.08	476	10.02	7.93	156
1233	903	3540	30	30	2	100	30.81	24.92	544	30.04	24.55	420
1235	903	3540	30	30	4	100	39.94	36.10	836	37.08	35.23	312
1237	903	3540	30	30	6	100	155.08	113.83	1613	143.78	111.96	559
1239	903	3540	30	30	8	100	149.24	96.78	576	137.10	95.40	154
1241	1203	4720	30	40	2	100	343.62	267.94	1276	313.32	260.41	797
1243	1203	4720	30	40	4	100	316.15	292.15	1132	302.56	283.07	419
1245	1203	4720	30	40	6	100	1088.94	615.35	1631	1000.14	639.58	476
1247	1203	4720	30	40	8	100	821.87	542.08	1763	805.75	525.14	501
1249	403	1580	40	10	2	100	0.11	0.07	75	0.10	0.08	54
1251	403	1580	40	10	4	100	0.19	0.21	81	0.19	0.20	34
1253	403	1580	40	10	6	100	0.32	0.32	75	0.32	0.33	44
1255	403	1580	40	10	8	100	0.27	0.29	86	0.26	0.30	34
1257	803	3160	40	20	2	100	1.77	1.24	272	1.77	1.24	220
1259	803	3160	40	20	4	100	7.20	5.62	248	6.40	5.63	134
1261	803	3160	40	20	6	100	5.50	5.28	209	5.42	5.16	110
1263	803	3160	40	20	8	100	9.62	8.17	571	9.52	10.02	153
1265	1203	4740	40	30	2	100	36.65	32.04	611	35.65	31.83	405
1267	1203	4740	40	30	4	100	65.64	65.62	861	61.00	65.43	293
1269	1203	4740	40	30	6	100	105.86	93.21	709	102.93	93.33	219
1271	1203	4740	40	30	8	100	261.07	179.21	1012	248.83	181.48	238
1273	1603	6320	40	40	2	100	437.69	429.50	1693	499.49	420.51	1044
1275	1603	6320	40	40	4	100	359.67	313.99	967	365.49	325.76	407
1277	1603	6320	40	40	6	100	1692.17	1011.59	2367	1746.84	1008.35	546
1279	1603	6320	40	40	8	100	1821.40	1340.75	2584	1872.05	1377.21	576

https://doi.org/10.1002/net.21815

TABLE A7 Grid instances with four objectives ($k = 4$), correlated scenarios and $c = 10$

	|*V*|	|*E*|	*h*	*w*	*r*	*c*	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
2050	103	380	10	10	2	10	0.08	0.07	145	0.08	0.04	52
2052	103	380	10	10	4	10	0.16	0.06	196	0.15	0.05	19
2054	203	760	10	20	2	10	1.39	0.47	409	1.29	0.50	72
2056	203	760	10	20	4	10	11.37	2.27	748	11.36	2.04	43
2058	303	1140	10	30	2	10	44.86	14.18	1857	47.83	13.19	251
2060	303	1140	10	30	4	10	375.28	36.62	3020	372.86	36.48	60
2062	203	780	20	10	2	10	0.09	0.04	76	0.09	0.04	22
2064	203	780	20	10	4	10	0.42	0.13	171	0.41	0.13	8
2066	403	1560	20	20	2	10	6.97	2.34	687	6.91	2.28	129
2068	403	1560	20	20	4	10	37.94	4.96	1197	37.86	4.95	32
2070	603	2340	20	30	2	10	210.21	59.32	2557	202.54	59.25	347
2072	603	2340	20	30	4	10	3272.60	419.15	8058	3074.41	425.33	95
2074	303	1180	30	10	2	10	0.20	0.08	117	0.20	0.09	44
2076	303	1180	30	10	4	10	0.83	0.27	261	0.81	0.23	17
2078	603	2360	30	20	2	10	23.83	8.54	827	24.71	8.57	125
2080	603	2360	30	20	4	10	104.13	15.94	1544	103.93	16.18	44
2082	903	3540	30	30	2	10	261.00	81.94	1487	256.21	82.26	197
2084	903	3540	30	30	4	10	3538.64	471.57	5264	3411.47	470.48	56

TABLE A8 Grid instances with four objectives ($k = 4$), correlated scenarios and $c = 100$

	|*V*|	|*E*|	*h*	*w*	*r*	*c*	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
2086	103	380	10	10	2	100	0.07	0.05	145	0.07	0.05	128
2088	103	380	10	10	4	100	0.08	0.07	107	0.07	0.07	61
2090	203	760	10	20	2	100	4.81	3.67	740	4.85	3.72	544
2092	203	760	10	20	4	100	8.52	7.05	819	8.45	7.21	495
2094	303	1140	10	30	2	100	101.37	64.69	2272	105.14	63.72	1577
2096	303	1140	10	30	4	100	107.88	75.80	2212	103.24	75.33	907
2098	203	780	20	10	2	100	0.10	0.07	191	0.10	0.07	154
2100	203	780	20	10	4	100	0.24	0.20	193	0.21	0.19	130
2102	403	1560	20	20	2	100	17.22	11.57	1013	16.84	11.65	813
2104	403	1560	20	20	4	100	33.92	31.04	674	34.13	30.64	366
2106	603	2340	20	30	2	100	1114.05	645.15	3925	1057.09	654.00	2715
2108	603	2340	20	30	4	100	1242.92	983.20	5751	1342.84	1002.42	2630
2110	303	1180	30	10	2	100	0.21	0.14	149	0.21	0.15	126
2112	303	1180	30	10	4	100	0.57	0.53	238	0.58	0.53	148
2114	603	2360	30	20	2	100	12.58	10.32	911	12.87	10.37	711
2116	603	2360	30	20	4	100	71.08	60.27	1703	73.71	61.04	824
2118	903	3540	30	30	2	100	462.29	366.25	3619	501.12	366.60	2446
2120	903	3540	30	30	4	100	1762.66	1598.68	4091	1849.92	1587.75	1823

https://doi.org/10.1002/net.21815

TABLE A9 Grid instances with two objectives ($k = 2$), random scenarios and $c = 10$, flimsily robust efficiency, 20 instances for each set of parameters

							EL-Flimsily			RL-Flimsily			sols		
	$\|V\|$	$\|E\|$	h	w	r	c	min	max	avg	min	max	avg	min	max	avg
2	103	380	10	10	2	10	0.00	0.01	0.01	0.00	0.00	0.00	7	22	15.35
4	103	380	10	10	4	10	0.01	0.03	0.02	0.00	0.01	0.00	23	43	30.20
6	103	380	10	10	6	10	0.03	0.06	0.04	0.01	0.01	0.01	34	63	43.85
8	103	380	10	10	8	10	0.06	0.09	0.08	0.01	0.01	0.01	51	73	60.60
10	203	760	10	20	2	10	0.03	0.06	0.04	0.01	0.02	0.01	22	49	35.30
12	203	760	10	20	4	10	0.12	0.24	0.18	0.02	0.04	0.03	52	81	66.65
14	203	760	10	20	6	10	0.33	0.56	0.46	0.03	0.05	0.04	73	114	97.25
16	203	760	10	20	8	10	0.77	1.40	1.01	0.07	0.14	0.09	102	165	137.85
18	303	1140	10	30	2	10	0.07	0.25	0.15	0.03	0.12	0.07	31	84	53.35
20	303	1140	10	30	4	10	0.54	1.03	0.79	0.07	0.17	0.12	88	143	115.70
22	303	1140	10	30	6	10	1.42	3.16	2.19	0.10	0.24	0.17	133	193	161.20
24	303	1140	10	30	8	10	3.26	5.75	4.67	0.14	0.25	0.21	169	266	218.65
26	403	1520	10	40	2	10	0.25	0.65	0.43	0.07	0.18	0.11	48	117	76.35
28	403	1520	10	40	4	10	1.49	3.00	2.20	0.17	0.32	0.26	133	185	158.55
30	403	1520	10	40	6	10	4.39	9.03	6.83	0.30	0.54	0.44	177	291	239.75
32	403	1520	10	40	8	10	10.87	20.85	14.88	0.46	0.79	0.59	271	422	326.85
34	203	780	20	10	2	10	0.01	0.01	0.01	0.00	0.00	0.00	11	25	16.75
36	203	780	20	10	4	10	0.03	0.05	0.04	0.01	0.01	0.01	25	45	32.90
38	203	780	20	10	6	10	0.07	0.12	0.10	0.01	0.02	0.01	31	68	52.35
40	203	780	20	10	8	10	0.16	0.20	0.17	0.02	0.02	0.02	54	77	63.65
42	403	1560	20	20	2	10	0.06	0.12	0.09	0.02	0.04	0.02	27	47	36.95
44	403	1560	20	20	4	10	0.32	0.51	0.40	0.05	0.07	0.06	54	84	70.15
46	403	1560	20	20	6	10	0.76	1.33	1.09	0.07	0.11	0.09	89	125	105.10
48	403	1560	20	20	8	10	1.81	2.52	2.21	0.10	0.14	0.12	118	158	135.30
50	603	2340	20	30	2	10	0.26	0.63	0.38	0.06	0.17	0.10	34	93	61.50
52	603	2340	20	30	4	10	1.25	2.28	1.77	0.15	0.27	0.21	89	130	112.45
54	603	2340	20	30	6	10	3.97	6.87	5.14	0.28	0.46	0.35	142	203	176.25
56	603	2340	20	30	8	10	9.88	12.60	10.88	0.43	0.55	0.48	204	279	235.55
58	803	3120	20	40	2	10	0.58	1.39	1.05	0.15	0.40	0.26	58	109	83.90
60	803	3120	20	40	4	10	4.14	8.93	6.08	0.48	0.94	0.68	126	195	172.10
62	803	3120	20	40	6	10	12.39	24.54	17.70	0.77	1.41	1.09	206	310	254.70
64	803	3120	20	40	8	10	27.27	46.56	38.35	1.06	1.71	1.44	277	388	334.90
66	303	1180	30	10	2	10	0.01	0.02	0.02	0.00	0.01	0.01	11	26	17.40
68	303	1180	30	10	4	10	0.05	0.07	0.06	0.01	0.01	0.01	28	46	35.85
70	303	1180	30	10	6	10	0.13	0.17	0.15	0.02	0.02	0.02	42	64	54.90
72	303	1180	30	10	8	10	0.27	0.33	0.29	0.03	0.03	0.03	56	82	69.05
74	603	2360	30	20	2	10	0.10	0.18	0.13	0.03	0.05	0.04	26	61	35.60
76	603	2360	30	20	4	10	0.50	0.76	0.64	0.07	0.11	0.09	57	100	75.55
78	603	2360	30	20	6	10	1.48	1.97	1.68	0.12	0.16	0.14	93	131	111.15
80	603	2360	30	20	8	10	3.07	4.18	3.51	0.17	0.22	0.19	122	168	143.85
82	903	3540	30	30	2	10	0.40	0.79	0.60	0.11	0.20	0.15	45	71	56.90
84	903	3540	30	30	4	10	2.12	3.52	2.90	0.25	0.42	0.34	96	154	117.90
86	903	3540	30	30	6	10	6.59	10.13	8.42	0.45	0.68	0.57	133	216	177.50
88	903	3540	30	30	8	10	15.69	20.75	18.72	0.68	0.88	0.80	203	275	238.55
90	1203	4720	30	40	2	10	1.24	2.31	1.83	0.30	0.54	0.43	68	101	89.35
92	1203	4720	30	40	4	10	7.62	10.91	9.04	0.81	1.22	1.02	129	220	173.40
94	1203	4720	30	40	6	10	20.49	34.46	27.34	1.34	2.09	1.70	225	327	263.75
96	1203	4720	30	40	8	10	50.04	71.68	59.48	1.97	2.78	2.31	289	382	344.40
98	403	1580	40	10	2	10	0.02	0.03	0.02	0.01	0.01	0.01	10	30	18.80
100	403	1580	40	10	4	10	0.07	0.10	0.09	0.01	0.02	0.02	26	45	36.30
102	403	1580	40	10	6	10	0.18	0.23	0.21	0.02	0.03	0.03	41	71	55.75
104	403	1580	40	10	8	10	0.36	0.46	0.40	0.04	0.04	0.04	58	86	71.25
106	803	3160	40	20	2	10	0.15	0.24	0.20	0.04	0.07	0.05	27	45	36.45
108	803	3160	40	20	4	10	0.70	1.04	0.84	0.10	0.14	0.12	47	89	72.30
110	803	3160	40	20	6	10	2.00	2.67	2.35	0.17	0.21	0.19	73	144	111.60
112	803	3160	40	20	8	10	4.03	5.74	4.90	0.23	0.30	0.26	127	192	147.15
114	1203	4740	40	30	2	10	0.56	1.03	0.75	0.14	0.25	0.19	39	79	59.70
116	1203	4740	40	30	4	10	3.17	4.69	3.93	0.39	0.56	0.48	94	141	119.75
118	1203	4740	40	30	6	10	9.97	15.74	11.34	0.71	1.01	0.79	148	233	188.85
120	1203	4740	40	30	8	10	20.71	27.80	23.90	0.94	1.17	1.06	188	270	226.15
122	1603	6320	40	40	2	10	1.77	3.21	2.34	0.40	0.76	0.58	50	118	89.05
124	1603	6320	40	40	4	10	9.30	18.57	12.46	1.12	2.22	1.46	157	232	187.40
126	1603	6320	40	40	6	10	29.73	43.57	37.70	1.98	2.69	2.36	215	305	273.80
128	1603	6320	40	40	8	10	79.32	96.18	85.94	3.07	3.87	3.38	296	449	352.50

https://doi.org/10.1002/net.21815

TABLE A10 Grid instances with two objectives ($k = 2$), random scenarios and $c = 100$, flimsily robust efficiency, 20 instances for each set of parameters

							EL-Flimsily			RL-Flimsily			sols		
	$\|V\|$	$\|E\|$	h	w	r	c	min	max	avg	min	max	avg	min	max	avg
130	103	380	10	10	2	100	0.01	0.01	0.01	0.00	0.01	0.00	11	32	23.25
132	103	380	10	10	4	100	0.02	0.05	0.04	0.00	0.01	0.01	24	60	41.20
134	103	380	10	10	6	100	0.07	0.13	0.08	0.01	0.02	0.01	50	102	66.55
136	103	380	10	10	8	100	0.13	0.22	0.16	0.01	0.03	0.01	69	111	87.75
138	203	760	10	20	2	100	0.04	0.14	0.07	0.01	0.04	0.02	30	74	48.90
140	203	760	10	20	4	100	0.30	0.62	0.41	0.04	0.08	0.06	72	154	101.85
142	203	760	10	20	6	100	0.77	1.39	1.07	0.06	0.12	0.09	130	206	160.65
144	203	760	10	20	8	100	1.51	3.27	2.22	0.08	0.17	0.11	160	237	195.35
146	303	1140	10	30	2	100	0.24	0.69	0.36	0.06	0.18	0.09	56	135	87.95
148	303	1140	10	30	4	100	1.40	2.71	2.00	0.16	0.34	0.24	138	235	184.55
150	303	1140	10	30	6	100	4.25	7.88	5.61	0.29	0.53	0.36	233	311	267.15
152	303	1140	10	30	8	100	10.24	20.88	13.38	0.41	0.82	0.53	308	464	372.70
154	403	1520	10	40	2	100	0.48	1.87	1.32	0.15	0.49	0.34	91	199	143.85
156	403	1520	10	40	4	100	3.78	9.37	6.19	0.49	1.30	0.82	192	321	268.25
158	403	1520	10	40	6	100	13.52	37.13	21.72	0.77	2.86	1.23	337	532	407.90
160	403	1520	10	40	8	100	30.67	64.74	46.31	1.09	1.98	1.48	429	674	554.40
162	203	780	20	10	2	100	0.01	0.04	0.02	0.00	0.01	0.01	17	41	26.60
164	203	780	20	10	4	100	0.07	0.12	0.08	0.01	0.02	0.02	39	66	50.45
166	203	780	20	10	6	100	0.16	0.24	0.20	0.02	0.03	0.02	59	94	74.40
168	203	780	20	10	8	100	0.32	0.51	0.39	0.03	0.05	0.03	78	122	98.55
170	403	1560	20	20	2	100	0.12	0.37	0.21	0.04	0.09	0.06	39	90	59.25
172	403	1560	20	20	4	100	0.67	1.28	0.94	0.09	0.16	0.12	82	132	106.95
174	403	1560	20	20	6	100	1.79	3.16	2.52	0.16	0.25	0.20	143	200	168.75
176	403	1560	20	20	8	100	4.37	7.26	5.51	0.23	0.35	0.27	170	257	221.55
178	603	2340	20	30	2	100	0.51	1.23	0.84	0.14	0.30	0.22	66	118	92.90
180	603	2340	20	30	4	100	3.31	6.54	4.84	0.39	0.68	0.54	161	237	194.90
182	603	2340	20	30	6	100	11.78	18.09	13.88	0.67	1.04	0.83	234	356	280.90
184	603	2340	20	30	8	100	23.98	38.39	32.07	0.93	1.57	1.17	334	441	380.50
186	803	3120	20	40	2	100	2.24	4.33	2.93	0.46	1.06	0.76	113	203	142.55
188	803	3120	20	40	4	100	11.19	20.95	16.57	1.29	2.42	1.88	239	379	297.10
190	803	3120	20	40	6	100	41.37	73.48	55.11	2.16	3.75	2.94	343	530	443.30
192	803	3120	20	40	8	100	94.52	157.72	129.16	2.87	6.05	4.26	507	697	601.15
194	303	1180	30	10	2	100	0.03	0.05	0.03	0.01	0.02	0.01	19	36	27.35
196	303	1180	30	10	4	100	0.10	0.15	0.12	0.02	0.02	0.02	41	57	48.60
198	303	1180	30	10	6	100	0.29	0.39	0.33	0.03	0.05	0.04	60	100	75.05
200	303	1180	30	10	8	100	0.53	0.75	0.61	0.04	0.06	0.05	82	120	98.55
202	603	2360	30	20	2	100	0.23	0.43	0.33	0.06	0.11	0.09	38	75	58.45
204	603	2360	30	20	4	100	1.10	1.94	1.53	0.14	0.27	0.20	88	148	118.80
206	603	2360	30	20	6	100	3.29	5.44	4.27	0.25	0.39	0.30	127	253	175.45
208	603	2360	30	20	8	100	7.83	10.43	9.17	0.36	0.47	0.42	181	283	231.85
210	903	3540	30	30	2	100	0.90	1.95	1.44	0.23	0.50	0.37	68	158	96.70
212	903	3540	30	30	4	100	5.77	8.79	7.06	0.66	0.99	0.80	158	241	197.00
214	903	3540	30	30	6	100	16.89	28.39	23.41	1.03	1.62	1.36	231	326	279.30
216	903	3540	30	30	8	100	44.50	63.35	52.60	1.67	2.35	1.99	338	493	387.95
218	1203	4720	30	40	2	100	3.56	6.65	4.88	0.81	1.91	1.25	111	189	151.95
220	1203	4720	30	40	4	100	21.36	32.03	25.69	2.39	3.65	2.84	219	322	287.50
222	1203	4720	30	40	6	100	74.43	110.14	86.56	4.09	6.55	5.23	369	536	442.05
224	1203	4720	30	40	8	100	166.00	256.82	209.89	5.63	8.72	7.06	474	719	598.85
226	403	1580	40	10	2	100	0.03	0.06	0.04	0.01	0.02	0.01	15	46	26.90
228	403	1580	40	10	4	100	0.14	0.23	0.17	0.02	0.06	0.03	35	75	52.35
230	403	1580	40	10	6	100	0.36	0.50	0.43	0.04	0.07	0.05	55	97	76.10
232	403	1580	40	10	8	100	0.72	1.00	0.82	0.06	0.08	0.07	90	144	106.05
234	803	3160	40	20	2	100	0.27	0.48	0.40	0.07	0.16	0.12	33	83	57.90
236	803	3160	40	20	4	100	1.53	2.27	1.96	0.20	0.38	0.27	83	134	110.65
238	803	3160	40	20	6	100	4.64	6.75	5.61	0.34	0.54	0.42	147	205	174.25
240	803	3160	40	20	8	100	10.61	17.16	12.84	0.51	0.83	0.60	194	286	230.80
242	1203	4740	40	30	2	100	1.44	2.36	1.92	0.38	0.65	0.51	65	120	95.00
244	1203	4740	40	30	4	100	7.86	13.25	10.57	0.97	1.63	1.30	162	230	199.80
246	1203	4740	40	30	6	100	23.77	39.15	31.23	1.49	2.31	1.90	228	352	291.10
248	1203	4740	40	30	8	100	65.11	82.88	73.53	2.40	3.18	2.77	343	489	409.90
250	1603	6320	40	40	2	100	4.44	8.98	6.15	1.13	2.40	1.72	103	217	146.65
252	1603	6320	40	40	4	100	27.39	43.30	35.39	3.01	6.18	4.37	225	387	299.90
254	1603	6320	40	40	6	100	94.92	139.25	118.61	5.47	10.46	6.83	345	500	445.05
256	1603	6320	40	40	8	100	236.51	336.08	276.11	7.87	12.33	9.56	511	685	592.40

https://doi.org/10.1002/net.21815

TABLE A11 Grid instances with two objectives ($k = 2$), random scenarios and $c = 10$, highly robust efficiency, 20 instances for each set of parameters

	|V|	|E|	h	w	r	c	EL-Highly min	EL-Highly max	EL-Highly avg	RL-Highly min	RL-Highly max	RL-Highly avg	sols min	sols max	sols avg
2	103	380	10	10	2	10	0.00	0.01	0.01	0.00	0.00	0.00	0	1	0.05
4	103	380	10	10	4	10	0.01	0.03	0.02	0.00	0.01	0.00	0	0	0.00
6	103	380	10	10	6	10	0.03	0.06	0.04	0.00	0.01	0.01	0	0	0.00
8	103	380	10	10	8	10	0.06	0.10	0.08	0.01	0.01	0.01	0	0	0.00
10	203	760	10	20	2	10	0.03	0.06	0.04	0.01	0.02	0.01	0	1	0.05
12	203	760	10	20	4	10	0.12	0.25	0.18	0.02	0.03	0.03	0	0	0.00
14	203	760	10	20	6	10	0.34	0.56	0.46	0.03	0.05	0.04	0	0	0.00
16	203	760	10	20	8	10	0.80	1.37	1.02	0.05	0.07	0.06	0	0	0.00
18	303	1140	10	30	2	10	0.07	0.26	0.15	0.02	0.07	0.04	0	0	0.00
20	303	1140	10	30	4	10	0.55	1.02	0.80	0.07	0.13	0.10	0	0	0.00
22	303	1140	10	30	6	10	1.44	3.26	2.24	0.10	0.21	0.16	0	0	0.00
24	303	1140	10	30	8	10	3.04	5.91	4.70	0.14	0.24	0.20	0	0	0.00
26	403	1520	10	40	2	10	0.26	0.66	0.43	0.07	0.18	0.11	0	0	0.00
28	403	1520	10	40	4	10	1.51	2.95	2.25	0.17	0.31	0.25	0	0	0.00
30	403	1520	10	40	6	10	4.47	9.36	7.05	0.29	0.53	0.43	0	0	0.00
32	403	1520	10	40	8	10	10.91	21.87	15.39	0.45	0.79	0.58	0	0	0.00
34	203	780	20	10	2	10	0.01	0.01	0.01	0.00	0.00	0.00	0	1	0.05
36	203	780	20	10	4	10	0.03	0.05	0.04	0.01	0.01	0.01	0	0	0.00
38	203	780	20	10	6	10	0.07	0.12	0.10	0.01	0.02	0.01	0	0	0.00
40	203	780	20	10	8	10	0.16	0.20	0.18	0.02	0.02	0.02	0	0	0.00
42	403	1560	20	20	2	10	0.06	0.13	0.09	0.02	0.04	0.02	0	0	0.00
44	403	1560	20	20	4	10	0.32	0.51	0.40	0.05	0.07	0.06	0	0	0.00
46	403	1560	20	20	6	10	0.81	1.35	1.11	0.07	0.11	0.09	0	0	0.00
48	403	1560	20	20	8	10	1.83	2.69	2.27	0.10	0.14	0.12	0	0	0.00
50	603	2340	20	30	2	10	0.26	0.64	0.38	0.06	0.17	0.10	0	0	0.00
52	603	2340	20	30	4	10	1.34	2.28	1.82	0.15	0.27	0.21	0	0	0.00
54	603	2340	20	30	6	10	4.06	7.04	5.29	0.28	0.46	0.35	0	0	0.00
56	603	2340	20	30	8	10	9.96	12.92	11.13	0.43	0.55	0.48	0	0	0.00
58	803	3120	20	40	2	10	0.60	1.44	1.09	0.14	0.41	0.26	0	0	0.00
60	803	3120	20	40	4	10	4.29	9.33	6.27	0.47	0.95	0.68	0	0	0.00
62	803	3120	20	40	6	10	11.96	24.82	17.89	0.77	1.40	1.07	0	0	0.00
64	803	3120	20	40	8	10	28.02	47.04	38.95	1.06	1.69	1.43	0	0	0.00
66	303	1180	30	10	2	10	0.01	0.02	0.02	0.00	0.01	0.01	0	0	0.00
68	303	1180	30	10	4	10	0.06	0.08	0.07	0.01	0.01	0.01	0	0	0.00
70	303	1180	30	10	6	10	0.13	0.17	0.15	0.02	0.02	0.02	0	0	0.00
72	303	1180	30	10	8	10	0.27	0.33	0.29	0.03	0.03	0.03	0	0	0.00
74	603	2360	30	20	2	10	0.10	0.18	0.13	0.03	0.05	0.04	0	0	0.00
76	603	2360	30	20	4	10	0.50	0.76	0.65	0.07	0.10	0.09	0	0	0.00
78	603	2360	30	20	6	10	1.50	2.15	1.72	0.12	0.16	0.14	0	0	0.00
80	603	2360	30	20	8	10	3.16	4.31	3.58	0.17	0.22	0.19	0	0	0.00
82	903	3540	30	30	2	10	0.40	0.79	0.61	0.11	0.20	0.15	0	0	0.00
84	903	3540	30	30	4	10	2.18	3.45	2.94	0.25	0.41	0.34	0	0	0.00
86	903	3540	30	30	6	10	6.56	10.16	8.49	0.44	0.70	0.57	0	0	0.00
88	903	3540	30	30	8	10	15.88	21.04	18.99	0.68	1.01	0.81	0	0	0.00
90	1203	4720	30	40	2	10	1.23	2.29	1.85	0.30	0.53	0.44	0	0	0.00
92	1203	4720	30	40	4	10	7.60	10.96	9.08	0.80	1.19	1.02	0	0	0.00
94	1203	4720	30	40	6	10	20.80	35.20	27.99	1.32	2.09	1.69	0	0	0.00
96	1203	4720	30	40	8	10	51.11	72.96	60.48	2.01	2.77	2.30	0	0	0.00
98	403	1580	40	10	2	10	0.02	0.03	0.02	0.01	0.01	0.01	0	1	0.15
100	403	1580	40	10	4	10	0.07	0.10	0.09	0.01	0.02	0.02	0	0	0.00
102	403	1580	40	10	6	10	0.18	0.24	0.21	0.02	0.03	0.03	0	0	0.00
104	403	1580	40	10	8	10	0.36	0.51	0.41	0.03	0.04	0.04	0	0	0.00
106	803	3160	40	20	2	10	0.15	0.24	0.20	0.04	0.07	0.05	0	0	0.00
108	803	3160	40	20	4	10	0.73	1.08	0.86	0.10	0.14	0.12	0	0	0.00
110	803	3160	40	20	6	10	2.02	2.77	2.39	0.17	0.21	0.19	0	0	0.00
112	803	3160	40	20	8	10	4.11	5.83	5.06	0.23	0.29	0.26	0	0	0.00
114	1203	4740	40	30	2	10	0.58	1.04	0.76	0.14	0.25	0.19	0	0	0.00
116	1203	4740	40	30	4	10	3.26	4.79	3.98	0.40	0.56	0.48	0	0	0.00
118	1203	4740	40	30	6	10	10.24	16.01	11.52	0.71	0.99	0.79	0	0	0.00
120	1203	4740	40	30	8	10	21.25	28.06	24.39	0.91	1.14	1.05	0	0	0.00
122	1603	6320	40	40	2	10	1.73	3.46	2.36	0.40	0.75	0.58	0	0	0.00
124	1603	6320	40	40	4	10	9.44	18.74	12.66	1.11	2.22	1.46	0	0	0.00
126	1603	6320	40	40	6	10	30.90	44.19	38.33	2.13	2.73	2.38	0	0	0.00
128	1603	6320	40	40	8	10	80.07	97.28	87.13	2.94	3.77	3.34	0	0	0.00

https://doi.org/10.1002/net.21815

TABLE A12 Grid instances with two objectives ($k = 2$), random scenarios and $c = 100$, highly robust efficiency, 20 instances for each set of parameters

							EL-Highly			**RL-Highly**			**sols**		
	$\|V\|$	$\|E\|$	h	w	r	c	**min**	**max**	**avg**	**min**	**max**	**avg**	**min**	**max**	**avg**
130	103	380	10	10	2	100	0.01	0.02	0.01	0.00	0.01	0.00	0	1	0.05
132	103	380	10	10	4	100	0.02	0.06	0.04	0.00	0.01	0.01	0	0	0.00
134	103	380	10	10	6	100	0.06	0.13	0.08	0.01	0.02	0.01	0	0	0.00
136	103	380	10	10	8	100	0.13	0.23	0.17	0.01	0.02	0.01	0	0	0.00
138	203	760	10	20	2	100	0.04	0.15	0.07	0.01	0.05	0.02	0	1	0.05
140	203	760	10	20	4	100	0.30	0.62	0.41	0.04	0.08	0.06	0	0	0.00
142	203	760	10	20	6	100	0.77	1.46	1.08	0.06	0.11	0.08	0	0	0.00
144	203	760	10	20	8	100	1.51	3.28	2.23	0.08	0.18	0.11	0	0	0.00
146	303	1140	10	30	2	100	0.25	0.72	0.37	0.06	0.17	0.10	0	0	0.00
148	303	1140	10	30	4	100	1.38	2.71	2.00	0.17	0.32	0.25	0	0	0.00
150	303	1140	10	30	6	100	4.23	8.16	5.62	0.29	0.71	0.38	0	0	0.00
152	303	1140	10	30	8	100	10.04	21.22	13.50	0.41	0.83	0.52	0	0	0.00
154	403	1520	10	40	2	100	0.49	1.72	1.29	0.16	0.45	0.32	0	0	0.00
156	403	1520	10	40	4	100	3.85	9.14	6.22	0.43	0.99	0.69	0	0	0.00
158	403	1520	10	40	6	100	14.23	38.00	21.94	0.89	1.92	1.18	0	0	0.00
160	403	1520	10	40	8	100	31.76	64.35	48.03	1.12	2.03	1.52	0	0	0.00
162	203	780	20	10	2	100	0.01	0.03	0.02	0.00	0.01	0.01	0	1	0.05
164	203	780	20	10	4	100	0.06	0.13	0.09	0.01	0.02	0.01	0	0	0.00
166	203	780	20	10	6	100	0.16	0.23	0.20	0.02	0.03	0.02	0	0	0.00
168	203	780	20	10	8	100	0.33	0.53	0.39	0.03	0.04	0.03	0	0	0.00
170	403	1560	20	20	2	100	0.12	0.35	0.20	0.04	0.08	0.06	0	0	0.00
172	403	1560	20	20	4	100	0.66	1.23	0.91	0.10	0.17	0.12	0	0	0.00
174	403	1560	20	20	6	100	1.78	3.17	2.53	0.14	0.31	0.22	0	0	0.00
176	403	1560	20	20	8	100	4.43	7.14	5.53	0.23	0.36	0.28	0	0	0.00
178	603	2340	20	30	2	100	0.51	1.23	0.86	0.13	0.33	0.24	0	0	0.00
180	603	2340	20	30	4	100	3.27	6.58	4.88	0.37	0.70	0.54	0	0	0.00
182	603	2340	20	30	6	100	11.77	17.85	14.03	0.65	1.11	0.82	0	0	0.00
184	603	2340	20	30	8	100	25.08	41.91	33.67	0.90	1.44	1.16	0	0	0.00
186	803	3120	20	40	2	100	2.25	4.37	2.95	0.45	1.04	0.74	0	0	0.00
188	803	3120	20	40	4	100	11.33	21.13	16.81	1.28	2.08	1.67	0	0	0.00
190	803	3120	20	40	6	100	46.07	77.30	57.56	2.16	4.32	2.96	0	0	0.00
192	803	3120	20	40	8	100	91.43	151.13	131.65	2.85	5.03	4.01	0	0	0.00
194	303	1180	30	10	2	100	0.03	0.04	0.03	0.01	0.02	0.01	0	1	0.05
196	303	1180	30	10	4	100	0.10	0.16	0.12	0.02	0.03	0.02	0	0	0.00
198	303	1180	30	10	6	100	0.30	0.42	0.33	0.03	0.04	0.03	0	0	0.00
200	303	1180	30	10	8	100	0.53	0.78	0.62	0.04	0.06	0.05	0	0	0.00
202	603	2360	30	20	2	100	0.24	0.44	0.33	0.07	0.13	0.09	0	0	0.00
204	603	2360	30	20	4	100	1.13	1.97	1.54	0.16	0.26	0.21	0	0	0.00
206	603	2360	30	20	6	100	3.29	5.53	4.30	0.25	0.39	0.31	0	0	0.00
208	603	2360	30	20	8	100	7.85	10.73	9.26	0.36	0.48	0.42	0	0	0.00
210	903	3540	30	30	2	100	0.91	1.96	1.44	0.23	0.54	0.38	0	0	0.00
212	903	3540	30	30	4	100	5.91	8.89	7.13	0.67	0.96	0.81	0	0	0.00
214	903	3540	30	30	6	100	17.54	30.36	24.30	1.02	1.72	1.39	0	0	0.00
216	903	3540	30	30	8	100	44.71	67.51	55.43	1.60	2.92	2.06	0	0	0.00
218	1203	4720	30	40	2	100	3.71	6.72	4.93	0.79	1.92	1.29	0	0	0.00
220	1203	4720	30	40	4	100	21.55	35.97	27.13	2.32	3.35	2.83	0	0	0.00
222	1203	4720	30	40	6	100	75.70	115.70	89.83	3.93	7.95	5.38	0	0	0.00
224	1203	4720	30	40	8	100	164.84	285.93	215.87	5.51	9.89	7.52	0	0	0.00
226	403	1580	40	10	2	100	0.03	0.06	0.04	0.01	0.02	0.01	0	1	0.05
228	403	1580	40	10	4	100	0.14	0.23	0.18	0.02	0.04	0.03	0	0	0.00
230	403	1580	40	10	6	100	0.36	0.50	0.44	0.04	0.06	0.05	0	0	0.00
232	403	1580	40	10	8	100	0.75	1.06	0.84	0.05	0.11	0.07	0	0	0.00
234	803	3160	40	20	2	100	0.27	0.50	0.41	0.08	0.17	0.12	0	0	0.00
236	803	3160	40	20	4	100	1.55	2.31	1.99	0.20	0.31	0.27	0	0	0.00
238	803	3160	40	20	6	100	4.60	6.99	5.71	0.36	0.55	0.44	0	0	0.00
240	803	3160	40	20	8	100	10.88	17.21	13.15	0.51	0.90	0.66	0	0	0.00
242	1203	4740	40	30	2	100	1.47	2.43	1.96	0.38	0.72	0.50	0	0	0.00
244	1203	4740	40	30	4	100	8.13	14.39	10.90	0.90	1.45	1.19	0	0	0.00
246	1203	4740	40	30	6	100	23.58	39.68	31.96	1.47	2.22	1.84	0	0	0.00
248	1203	4740	40	30	8	100	65.23	84.04	75.04	2.35	2.97	2.64	0	0	0.00
250	1603	6320	40	40	2	100	4.45	8.06	6.20	1.08	2.13	1.51	0	0	0.00
252	1603	6320	40	40	4	100	27.58	44.04	36.00	3.02	4.83	3.84	0	0	0.00
254	1603	6320	40	40	6	100	98.10	147.79	122.18	5.42	7.37	6.26	0	0	0.00
256	1603	6320	40	40	8	100	244.55	343.90	278.67	7.45	10.67	8.72	0	0	0.00

https://doi.org/10.1002/net.21815

TABLE A13 Grid instances with three objectives ($k = 3$), random scenarios and $c = 10$

	$\|V\|$	$\|E\|$	h	w	r	c	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
1026	103	380	10	10	2	10	0.02	0.01	45	0.02	0.01	0
1028	103	380	10	10	4	10	0.08	0.02	93	0.08	0.01	0
1030	103	380	10	10	6	10	0.25	0.03	146	0.24	0.04	0
1032	103	380	10	10	8	10	0.44	0.04	186	0.43	0.04	0
1034	203	760	10	20	2	10	0.60	0.17	228	0.61	0.17	1
1036	203	760	10	20	4	10	2.29	0.25	401	2.25	0.26	0
1038	203	760	10	20	6	10	8.81	0.57	496	8.43	0.51	0
1040	203	760	10	20	8	10	17.28	0.64	781	17.37	0.63	0
1042	303	1140	10	30	2	10	5.51	1.47	588	5.36	1.47	1
1044	303	1140	10	30	4	10	33.68	2.70	932	27.06	2.68	0
1046	303	1140	10	30	6	10	106.44	4.05	1220	104.26	4.05	0
1048	303	1140	10	30	8	10	242.27	5.51	1642	237.83	5.48	0
1050	403	1520	10	40	2	10	9.20	2.32	718	9.36	2.32	0
1052	403	1520	10	40	4	10	148.36	10.65	1847	145.65	10.67	0
1054	403	1520	10	40	6	10	491.12	16.51	2204	483.56	16.21	0
1056	403	1520	10	40	8	10	1234.74	32.54	3295	1198.43	28.95	0
1058	203	780	20	10	2	10	0.05	0.02	46	0.05	0.02	0
1060	203	780	20	10	4	10	0.24	0.05	133	0.24	0.06	0
1062	203	780	20	10	6	10	0.57	0.07	230	0.55	0.06	0
1064	203	780	20	10	8	10	0.89	0.08	200	0.89	0.07	0
1066	403	1560	20	20	2	10	1.31	0.40	205	1.28	0.32	0
1068	403	1560	20	20	4	10	6.64	0.90	434	6.76	0.73	0
1070	403	1560	20	20	6	10	20.80	1.26	679	21.07	1.13	0
1072	403	1560	20	20	8	10	44.36	1.60	818	43.00	1.42	0
1074	603	2340	20	30	2	10	14.66	3.71	610	13.91	3.74	0
1076	603	2340	20	30	4	10	74.66	5.88	928	66.94	5.96	0
1078	603	2340	20	30	6	10	352.76	16.43	1415	315.07	12.62	0
1080	603	2340	20	30	8	10	568.69	13.23	1798	527.84	12.80	0
1082	803	3120	20	40	2	10	111.45	30.58	1150	91.68	25.05	0
1084	803	3120	20	40	4	10	418.06	31.76	1839	357.96	28.81	0
1086	803	3120	20	40	6	10	1952.35	79.62	2729	1636.52	68.01	0
1088	803	3120	20	40	8	10	> 3600.00	97.92	4023	3512.34	85.67	0
1090	303	1180	30	10	2	10	0.11	0.03	59	0.09	0.03	0
1092	303	1180	30	10	4	10	0.25	0.04	92	0.25	0.04	0
1094	303	1180	30	10	6	10	0.80	0.09	165	0.80	0.08	0
1096	303	1180	30	10	8	10	1.94	0.14	304	1.89	0.13	0
1098	603	2360	30	20	2	10	3.25	0.68	311	2.77	0.69	0
1100	603	2360	30	20	4	10	12.44	1.20	395	10.93	1.19	0
1102	603	2360	30	20	6	10	38.77	1.76	644	33.41	1.83	0
1104	603	2360	30	20	8	10	83.17	2.54	760	77.84	2.47	0
1106	903	3540	30	30	2	10	16.44	4.97	530	16.46	4.30	0
1108	903	3540	30	30	4	10	132.24	15.91	1033	129.25	13.25	0
1110	903	3540	30	30	6	10	488.53	25.34	1733	504.64	21.43	0
1112	903	3540	30	30	8	10	813.58	28.53	2040	868.63	23.06	0
1114	1203	4720	30	40	2	10	100.13	39.45	959	105.10	33.00	0
1116	1203	4720	30	40	4	10	665.71	82.27	2203	765.86	73.64	0
1118	1203	4720	30	40	6	10	> 3600.00	230.81	3432	> 3600.00	202.97	0
1120	1203	4720	30	40	8	10	> 3600.00	219.66	4489	> 3600.00	195.73	0
1122	403	1580	40	10	2	10	0.13	0.04	75	0.12	0.05	0
1124	403	1580	40	10	4	10	0.47	0.07	158	0.48	0.07	0
1126	403	1580	40	10	6	10	1.23	0.12	181	1.26	0.12	0
1128	403	1580	40	10	8	10	2.30	0.16	295	2.39	0.16	0
1130	803	3160	40	20	2	10	2.61	0.88	219	2.67	0.73	2
1132	803	3160	40	20	4	10	14.67	2.08	540	15.43	1.70	0
1134	803	3160	40	20	6	10	56.70	3.25	951	60.93	3.14	0
1136	803	3160	40	20	8	10	92.92	3.49	1014	100.45	3.15	0
1138	1203	4740	40	30	2	10	18.61	5.29	495	19.57	5.31	0
1140	1203	4740	40	30	4	10	188.02	23.59	1347	206.80	20.02	0
1142	1203	4740	40	30	6	10	489.84	29.04	1547	518.47	24.07	0
1144	1203	4740	40	30	8	10	1165.91	37.14	2020	1256.95	35.74	0
1146	1603	6320	40	40	2	10	99.04	40.55	943	102.94	32.93	0
1148	1603	6320	40	40	4	10	1093.01	141.50	2035	1185.97	120.75	0
1150	1603	6320	40	40	6	10	> 3600.00	256.02	3829	> 3600.00	238.18	0
1152	1603	6320	40	40	8	10	> 3600.00	264.21	3624	> 3600.00	246.88	0

https://doi.org/10.1002/net.21815

TABLE A14 Grid instances with three objectives ($k = 3$), random scenarios and $c = 100$

	$\|V\|$	$\|E\|$	h	w	r	c	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
1154	103	380	10	10	2	100	0.04	0.01	72	0.04	0.01	3
1156	103	380	10	10	4	100	0.23	0.04	177	0.24	0.04	0
1158	103	380	10	10	6	100	0.56	0.05	283	0.56	0.05	0
1160	103	380	10	10	8	100	0.81	0.06	309	0.86	0.07	0
1162	203	760	10	20	2	100	1.35	0.39	337	1.47	0.39	6
1164	203	760	10	20	4	100	7.87	0.81	640	8.95	0.81	0
1166	203	760	10	20	6	100	26.56	1.39	952	34.07	1.38	0
1168	203	760	10	20	8	100	63.08	1.98	1272	74.09	1.95	0
1170	303	1140	10	30	2	100	20.27	3.87	716	22.95	3.82	0
1172	303	1140	10	30	4	100	66.37	5.58	1388	77.13	5.55	0
1174	303	1140	10	30	6	100	684.83	22.33	2702	765.73	22.19	0
1176	303	1140	10	30	8	100	873.13	21.27	4133	1012.57	20.86	0
1178	403	1520	10	40	2	100	146.11	38.21	2300	165.53	37.03	0
1180	403	1520	10	40	4	100	775.24	68.38	3471	901.80	68.28	0
1182	403	1520	10	40	6	100	2506.19	90.50	4589	2823.64	112.26	0
1184	403	1520	10	40	8	100	> 3600.00	142.47	6759	> 3600.00	135.13	0
1186	203	780	20	10	2	100	0.14	0.04	109	0.13	0.05	0
1188	203	780	20	10	4	100	0.57	0.08	205	0.56	0.08	0
1190	203	780	20	10	6	100	1.30	0.14	303	1.29	0.13	0
1192	203	780	20	10	8	100	2.91	0.21	394	2.91	0.18	0
1194	403	1560	20	20	2	100	2.07	0.62	273	2.07	0.56	0
1196	403	1560	20	20	4	100	16.12	2.21	648	16.34	1.68	0
1198	403	1560	20	20	6	100	73.50	5.42	1266	73.70	3.17	0
1200	403	1560	20	20	8	100	244.92	6.70	1735	222.19	5.99	0
1202	603	2340	20	30	2	100	55.67	16.18	1163	46.58	13.73	0
1204	603	2340	20	30	4	100	333.06	30.67	1882	282.90	26.70	0
1206	603	2340	20	30	6	100	1620.06	63.09	3099	1436.10	58.27	0
1208	603	2340	20	30	8	100	2055.06	53.99	3228	1843.03	49.93	0
1210	803	3120	20	40	2	100	550.67	143.66	2839	521.30	138.06	0
1212	803	3120	20	40	4	100	2426.44	262.89	4411	2358.97	256.39	0
1214	803	3120	20	40	6	100	> 3600.00	360.39	6602	> 3600.00	340.70	0
1216	803	3120	20	40	8	100	> 3600.00	622.66	7463	> 3600.00	585.15	0
1218	303	1180	30	10	2	100	0.19	0.05	115	0.17	0.05	3
1220	303	1180	30	10	4	100	0.67	0.10	179	0.66	0.10	0
1222	303	1180	30	10	6	100	2.23	0.19	282	2.14	0.18	0
1224	303	1180	30	10	8	100	4.05	0.25	418	4.00	0.25	0
1226	603	2360	30	20	2	100	5.04	1.33	421	4.96	1.29	1
1228	603	2360	30	20	4	100	40.44	3.81	1008	39.30	3.70	0
1230	603	2360	30	20	6	100	122.85	5.67	1311	118.17	5.60	0
1232	603	2360	30	20	8	100	328.83	8.38	1544	308.09	8.15	0
1234	903	3540	30	30	2	100	77.84	24.07	901	76.46	23.83	0
1236	903	3540	30	30	4	100	649.61	63.32	2138	615.86	62.24	0
1238	903	3540	30	30	6	100	2100.36	91.28	3188	1992.99	89.51	0
1240	903	3540	30	30	8	100	> 3600.00	147.84	4534	> 3600.00	144.45	0
1242	1203	4720	30	40	2	100	915.71	246.27	2450	870.75	239.12	0
1244	1203	4720	30	40	4	100	> 3600.00	438.21	4175	> 3600.00	427.43	0
1246	1203	4720	30	40	6	100	> 3600.00	788.21	7070	> 3600.00	764.83	0
1248	1203	4720	30	40	8	100	> 3600.00	818.04	7736	> 3600.00	816.31	0
1250	403	1580	40	10	2	100	0.24	0.07	102	0.24	0.07	2
1252	403	1580	40	10	4	100	1.22	0.17	221	1.14	0.17	0
1254	403	1580	40	10	6	100	3.92	0.25	342	3.37	0.25	0
1256	403	1580	40	10	8	100	7.59	0.40	527	7.10	0.38	0
1258	803	3160	40	20	2	100	12.08	2.42	480	9.08	2.42	1
1260	803	3160	40	20	4	100	57.09	5.22	1191	52.62	5.10	0
1262	803	3160	40	20	6	100	195.43	9.23	1377	194.14	9.05	0
1264	803	3160	40	20	8	100	424.89	11.31	2014	405.64	11.84	0
1266	1203	4740	40	30	2	100	98.15	31.98	842	94.77	31.18	0
1268	1203	4740	40	30	4	100	949.93	95.71	2942	895.28	96.66	0
1270	1203	4740	40	30	6	100	3143.94	145.54	3410	2893.93	145.00	0
1272	1203	4740	40	30	8	100	> 3600.00	177.09	4526	> 3600.00	175.63	0
1274	1603	6320	40	40	2	100	713.25	217.56	1957	771.89	215.14	0
1276	1603	6320	40	40	4	100	> 3600.00	689.56	4485	> 3600.00	693.92	0
1278	1603	6320	40	40	6	100	> 3600.00	858.13	6418	> 3600.00	852.94	0
1280	1603	6320	40	40	8	100	> 3600.00	1369.15	8956	> 3600.00	1251.78	0

https://doi.org/10.1002/net.21815

TABLE A15 Grid instances with four objectives ($k = 4$), random scenarios and $c = 10$

	$\|V\|$	$\|E\|$	h	w	r	c	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
2051	103	380	10	10	2	10	0.06	0.02	85	0.06	0.02	4
2053	103	380	10	10	4	10	0.32	0.05	261	0.30	0.04	0
2055	203	760	10	20	2	10	4.57	1.26	702	4.60	1.18	13
2057	203	760	10	20	4	10	18.24	1.69	1200	17.84	1.62	0
2059	303	1140	10	30	2	10	63.72	13.95	2250	64.74	13.87	3
2061	303	1140	10	30	4	10	704.48	47.53	4365	713.43	47.02	0
2063	203	780	20	10	2	10	0.19	0.05	182	0.19	0.06	12
2065	203	780	20	10	4	10	0.83	0.11	274	0.79	0.11	0
2067	403	1560	20	20	2	10	9.31	2.26	826	9.42	2.21	4
2069	403	1560	20	20	4	10	71.01	5.85	1318	72.11	5.83	0
2071	603	2340	20	30	2	10	242.44	50.55	2902	239.59	50.36	0
2073	603	2340	20	30	4	10	2188.69	179.08	6114	2108.02	175.02	0
2075	303	1180	30	10	2	10	0.35	0.09	207	0.34	0.09	10
2077	303	1180	30	10	4	10	1.29	0.18	356	1.28	0.18	1
2079	603	2360	30	20	2	10	10.27	2.61	559	10.33	2.56	0
2081	603	2360	30	20	4	10	118.85	9.88	1851	117.38	10.14	0
2083	903	3540	30	30	2	10	679.99	164.50	3121	657.14	164.40	2
2085	903	3540	30	30	4	10	2468.86	205.38	3757	2392.72	207.28	0

TABLE A16 Grid instances with four objectives ($k = 4$), random scenarios and $c = 100$

	$\|V\|$	$\|E\|$	h	w	r	c	EL-Flimsily	RL-Flimsily	sols	EL-Highly	RL-Highly	sols
2087	103	380	10	10	2	100	0.23	0.06	211	0.22	0.06	15
2089	103	380	10	10	4	100	0.98	0.14	482	0.97	0.13	2
2091	203	760	10	20	2	100	20.17	4.54	1143	19.46	4.45	15
2093	203	760	10	20	4	100	138.79	10.59	2423	141.79	10.37	1
2095	303	1140	10	30	2	100	339.03	68.72	4259	339.92	68.50	0
2097	303	1140	10	30	4	100	3285.91	264.25	9803	3225.54	268.28	0
2099	203	780	20	10	2	100	0.48	0.12	284	0.45	0.12	7
2101	203	780	20	10	4	100	2.56	0.30	528	2.51	0.29	0
2103	403	1560	20	20	2	100	59.49	13.63	1585	57.25	13.60	6
2105	403	1560	20	20	4	100	407.55	29.63	2987	391.00	29.57	0
2107	603	2340	20	30	2	100	1292.27	307.90	4996	1323.37	294.51	1
2109	603	2340	20	30	4	100	> 3600.00	723.44	9834	> 3600.00	724.34	0
2111	303	1180	30	10	2	100	0.93	0.21	347	0.81	0.21	14
2113	303	1180	30	10	4	100	4.17	0.50	763	4.22	0.47	0
2115	603	2360	30	20	2	100	90.71	24.55	1861	96.69	23.35	0
2117	603	2360	30	20	4	100	361.71	31.64	3139	397.64	31.66	0
2119	903	3540	30	30	2	100	1843.49	500.46	2588	1911.63	480.27	0
2121	903	3540	30	30	4	100	> 3600.00	1593.80	12188	> 3600.00	1585.40	0

https://doi.org/10.1002/net.21815

TABLE A17 NetMaker instances, flimsily robust efficiency, $k = 2$, correlated scenarios

	$\|V\|$	$\|E\|$	r	EL-Flimsily min	EL-Flimsily max	EL-Flimsily avg	RL-Flimsily min	RL-Flimsily max	RL-Flimsily avg	sols min	sols max	sols avg
1	101	450.80	2	0.00	0.01	0.01	0.00	0.01	0.00	2	13	5.35
5	101	445.60	4	0.00	0.02	0.01	0.00	0.01	0.01	2	41	9.50
9	101	446.35	6	0.01	0.04	0.01	0.01	0.03	0.01	1	31	10.60
13	101	443.10	8	0.01	0.03	0.01	0.00	0.03	0.01	1	17	6.40
17	201	901.25	2	0.01	0.02	0.01	0.00	0.02	0.01	1	25	8.35
21	201	901.60	4	0.01	0.05	0.02	0.01	0.03	0.02	1	34	10.10
25	201	906.20	6	0.02	0.09	0.04	0.01	0.06	0.03	1	34	11.40
29	201	894.95	8	0.02	0.16	0.05	0.01	0.10	0.04	1	22	9.55
33	401	1803.15	2	0.02	0.08	0.04	0.01	0.06	0.03	1	17	9.00
37	401	1799.75	4	0.03	0.21	0.08	0.02	0.16	0.06	3	65	16.10
41	401	1796.35	6	0.04	0.11	0.08	0.03	0.10	0.07	1	28	13.80
45	401	1792.05	8	0.04	0.23	0.12	0.04	0.21	0.10	2	35	13.05
49	801	3604.65	2	0.05	0.19	0.10	0.04	0.14	0.07	3	27	11.25
53	801	3605.00	4	0.10	0.48	0.19	0.07	0.40	0.15	2	79	14.90
57	801	3596.20	6	0.13	0.76	0.27	0.11	0.50	0.20	1	38	9.25
61	801	3591.75	8	0.19	1.32	0.50	0.14	0.51	0.29	4	43	13.70
65	1201	5391.95	2	0.10	0.31	0.17	0.06	0.27	0.13	2	28	11.75
69	1201	5415.70	4	0.12	0.73	0.32	0.10	0.54	0.24	1	50	14.00
73	1201	5414.80	6	0.28	0.94	0.62	0.19	0.54	0.38	3	49	15.30
77	1201	5413.30	8	0.37	1.64	0.70	0.29	0.91	0.45	1	40	14.65
81	1601	7236.00	2	0.17	0.52	0.30	0.10	0.27	0.17	1	26	12.90
85	1601	7188.50	4	0.27	0.75	0.43	0.19	0.51	0.29	1	66	16.70
89	1601	7178.65	6	0.37	1.36	0.82	0.32	0.81	0.52	2	48	16.25
93	1601	7212.30	8	0.49	2.00	1.11	0.41	1.19	0.75	4	42	19.50

TABLE A18 NetMaker instances, flimsily robust efficiency, $k = 3$, correlated scenarios

	$\|V\|$	$\|E\|$	r	EL-Flimsily min	EL-Flimsily max	EL-Flimsily avg	RL-Flimsily min	RL-Flimsily max	RL-Flimsily avg	sols min	sols max	sols avg
2	101	446.90	2	0.04	0.62	0.30	0.03	0.53	0.26	22	134	65.50
6	101	450.50	4	0.24	1.92	0.65	0.26	1.12	0.57	27	276	92.80
10	101	445.50	6	0.15	2.67	1.08	0.23	2.95	1.19	19	422	169.60
14	101	447.00	8	0.27	2.12	0.85	0.34	2.29	1.00	33	213	95.50
18	201	913.20	2	0.36	2.90	1.41	0.33	2.13	1.09	45	266	140.40
22	201	892.60	4	0.31	4.65	1.93	0.43	4.11	1.93	81	322	146.50
26	201	906.70	6	0.68	6.62	2.59	0.90	5.76	2.72	10	239	99.90
30	201	908.70	8	0.54	10.87	3.96	0.63	9.05	3.88	49	415	144.50
34	401	1818.60	2	2.01	8.78	4.44	1.29	7.30	3.30	22	273	136.40
38	401	1781.60	4	1.95	18.87	8.83	1.91	16.93	8.36	14	354	153.80
42	401	1784.90	6	4.22	16.29	8.80	4.04	15.25	8.63	28	412	171.50
46	401	1813.50	8	2.94	45.83	17.45	3.69	38.52	17.21	32	182	110.20
50	801	3615.00	2	5.58	64.93	20.88	5.00	53.73	17.87	5	455	112.70
54	801	3590.20	4	11.10	52.23	29.29	13.65	58.51	33.49	39	385	219.60
58	801	3611.40	6	24.79	84.01	44.97	25.10	86.58	49.04	81	554	243.40
62	801	3604.00	8	23.66	122.16	71.22	28.79	156.65	89.46	14	426	235.00
66	1201	5399.90	2	14.66	68.65	34.96	14.53	61.97	33.86	33	322	196.60
70	1201	5390.20	4	25.41	294.71	98.39	34.90	373.71	129.72	62	675	337.40
74	1201	5422.00	6	43.57	240.33	107.85	61.74	288.51	144.40	68	1207	335.00
78	1201	5406.00	8	33.10	262.59	134.06	50.84	397.31	196.54	60	1229	479.40

https://doi.org/10.1002/net.21815

TABLE A19 NetMaker instances, highly robust efficiency, $k = 2$, correlated scenarios

	$\|V\|$	$\|E\|$	r	EL-Highly min	EL-Highly max	EL-Highly avg	RL-Highly min	RL-Highly max	RL-Highly avg	sols min	sols max	sols avg
1	101	450.80	2	0.00	0.01	0.01	0.00	0.01	0.00	2	10	4.40
5	101	445.60	4	0.00	0.02	0.01	0.00	0.01	0.01	2	10	4.75
9	101	446.35	6	0.01	0.04	0.01	0.01	0.03	0.01	1	12	4.85
13	101	443.10	8	0.01	0.03	0.01	0.00	0.02	0.01	1	9	3.75
17	201	901.25	2	0.01	0.02	0.01	0.00	0.02	0.01	1	18	6.25
21	201	901.60	4	0.01	0.05	0.02	0.01	0.04	0.02	1	13	6.15
25	201	906.20	6	0.02	0.10	0.04	0.01	0.06	0.03	1	13	5.00
29	201	894.95	8	0.02	0.16	0.05	0.01	0.11	0.04	1	8	3.95
33	401	1803.15	2	0.02	0.08	0.04	0.01	0.06	0.03	1	14	7.30
37	401	1799.75	4	0.03	0.21	0.08	0.02	0.17	0.06	2	14	6.45
41	401	1796.35	6	0.04	0.11	0.08	0.03	0.11	0.06	1	12	5.80
45	401	1792.05	8	0.04	0.24	0.11	0.04	0.17	0.09	1	13	4.80
49	801	3604.65	2	0.05	0.19	0.10	0.04	0.12	0.07	2	22	8.30
53	801	3605.00	4	0.10	0.46	0.19	0.08	0.36	0.15	1	22	6.10
57	801	3596.20	6	0.14	0.75	0.27	0.10	0.46	0.20	1	8	3.60
61	801	3591.75	8	0.19	1.31	0.49	0.15	0.52	0.30	0	11	5.15
65	1201	5391.95	2	0.10	0.31	0.17	0.06	0.22	0.11	2	19	7.90
69	1201	5415.70	4	0.13	0.73	0.32	0.10	0.42	0.21	1	15	6.50
73	1201	5414.80	6	0.30	0.99	0.66	0.19	0.55	0.38	2	18	6.05
77	1201	5413.30	8	0.37	1.60	0.72	0.30	0.93	0.48	1	9	5.15
81	1601	7236.00	2	0.15	0.50	0.28	0.10	0.26	0.17	1	18	9.20
85	1601	7188.50	4	0.27	0.69	0.42	0.20	0.53	0.34	1	15	6.90
89	1601	7178.65	6	0.37	1.35	0.81	0.32	0.94	0.57	2	16	5.70
93	1601	7212.30	8	0.52	2.07	1.13	0.41	1.20	0.77	3	16	7.00

TABLE A20 NetMaker instances, highly robust efficiency, $k = 3$, correlated scenarios

	$\|V\|$	$\|E\|$	r	EL-Highly min	EL-Highly max	EL-Highly avg	RL-Highly min	RL-Highly max	RL-Highly avg	sols min	sols max	sols avg
2	101	446.90	2	0.04	0.64	0.32	0.03	0.53	0.26	19	125	58.40
6	101	450.50	4	0.27	2.06	0.69	0.26	1.15	0.58	23	208	70.10
10	101	445.50	6	0.16	2.71	1.11	0.23	2.97	1.21	19	230	102.00
14	101	447.00	8	0.27	2.21	0.86	0.35	2.30	1.02	15	130	62.70
18	201	913.20	2	0.36	2.78	1.39	0.32	2.10	1.09	40	212	117.60
22	201	892.60	4	0.36	4.78	1.98	0.42	4.08	1.94	54	205	103.30
26	201	906.70	6	0.72	6.30	2.54	0.89	5.77	2.59	8	163	68.30
30	201	908.70	8	0.55	11.45	4.23	0.64	8.89	3.88	32	219	81.00
34	401	1818.60	2	1.90	9.18	4.44	1.32	7.09	3.32	19	220	117.60
38	401	1781.60	4	1.96	18.64	8.95	1.95	17.04	8.28	11	248	101.90
42	401	1784.90	6	4.68	17.15	9.49	4.11	15.19	8.57	16	215	108.30
46	401	1813.50	8	2.96	47.23	17.73	3.53	38.51	17.07	22	122	66.90
50	801	3615.00	2	5.54	74.58	22.21	5.08	52.30	17.42	5	352	92.80
54	801	3590.20	4	14.53	58.07	33.23	13.54	59.83	34.42	34	271	149.70
58	801	3611.40	6	22.18	77.03	40.31	25.02	86.67	48.86	54	347	150.50
62	801	3604.00	8	21.40	117.46	64.12	30.24	159.05	89.44	8	222	129.20
66	1201	5399.90	2	14.94	83.02	37.02	14.74	60.38	33.47	31	285	165.30
70	1201	5390.20	4	26.12	303.98	100.14	36.24	364.93	125.59	37	448	219.90
74	1201	5422.00	6	44.32	242.01	110.85	59.54	288.50	140.93	49	615	180.40
78	1201	5406.00	8	42.21	282.48	146.08	55.20	391.82	193.26	34	582	230.90

https://doi.org/10.1002/net.21815

TABLE A21 NetMaker instances, flimsily robust efficiency, $k = 2$, random scenarios

	$\|V\|$	$\|E\|$	r	EL-Flimsily min	EL-Flimsily max	EL-Flimsily avg	RL-Flimsily min	RL-Flimsily max	RL-Flimsily avg	sols min	sols max	sols avg
3	101	444.95	2	0.00	0.03	0.01	0.00	0.01	0.00	1	25	10.05
7	101	448.35	4	0.02	0.06	0.03	0.01	0.02	0.01	5	36	19.60
11	101	449.35	6	0.04	0.14	0.08	0.01	0.02	0.01	9	56	26.35
15	101	447.60	8	0.07	0.21	0.11	0.01	0.03	0.02	14	80	34.25
19	201	908.60	2	0.02	0.04	0.03	0.01	0.02	0.01	1	42	16.65
23	201	897.05	4	0.04	0.16	0.09	0.01	0.03	0.02	8	52	23.75
27	201	901.05	6	0.10	0.33	0.20	0.02	0.04	0.03	10	74	35.65
31	201	887.20	8	0.21	0.54	0.35	0.03	0.05	0.04	12	93	48.75
35	401	1798.25	2	0.04	0.14	0.08	0.01	0.05	0.03	1	42	15.75
39	401	1798.15	4	0.19	0.43	0.28	0.04	0.08	0.06	7	48	26.35
43	401	1790.20	6	0.39	2.04	0.71	0.05	0.18	0.08	11	101	42.75
47	401	1795.75	8	0.67	2.98	1.57	0.06	0.18	0.12	18	90	49.50
51	801	3617.85	2	0.13	0.48	0.22	0.04	0.15	0.07	2	46	18.55
55	801	3621.45	4	0.64	1.28	0.95	0.10	0.20	0.14	9	80	35.80
59	801	3605.35	6	1.27	3.59	2.43	0.13	0.28	0.21	11	92	46.95
63	801	3618.15	8	2.59	7.72	5.12	0.18	0.38	0.30	7	119	58.00
67	1201	5393.00	2	0.21	0.59	0.43	0.07	0.21	0.15	4	39	19.05
71	1201	5406.85	4	1.23	2.63	1.88	0.18	0.32	0.25	8	52	37.85
75	1201	5394.40	6	2.79	9.22	4.83	0.25	0.59	0.38	13	103	43.20
79	1201	5408.90	8	5.70	16.11	9.58	0.38	0.81	0.53	35	154	80.60
83	1601	7188.15	2	0.43	1.06	0.66	0.12	0.25	0.17	4	36	17.15
87	1601	7240.00	4	1.88	3.68	2.59	0.27	0.49	0.35	8	86	41.85
91	1601	7196.85	6	4.15	10.78	7.25	0.38	0.76	0.55	15	93	54.30
95	1601	7184.00	8	7.01	19.52	13.43	0.51	1.09	0.81	23	156	67.90

TABLE A22 NetMaker instances, flimsily robust efficiency, $k = 3$, random scenarios

	$\|V\|$	$\|E\|$	r	EL-Flimsily min	EL-Flimsily max	EL-Flimsily avg	RL-Flimsily min	RL-Flimsily max	RL-Flimsily avg	sols min	sols max	sols avg
4	101	453.40	2	0.18	1.46	0.74	0.07	0.52	0.28	49	304	193.10
8	101	449.40	4	0.48	18.63	4.08	0.14	2.74	0.73	71	417	250.40
12	101	437.10	6	1.41	7.05	3.15	0.26	1.04	0.51	157	408	253.20
16	101	453.90	8	4.77	61.39	16.52	0.68	4.08	1.50	220	1613	582.60
20	201	900.30	2	0.51	6.29	2.88	0.23	1.53	0.88	92	400	231.50
24	201	896.10	4	3.73	21.87	10.32	0.84	3.77	1.87	77	725	295.50
28	201	896.50	6	7.29	51.11	24.63	1.15	5.11	2.67	97	647	387.70
32	201	903.70	8	11.55	134.84	63.47	1.18	9.03	4.79	63	1065	504.50
36	401	1805.20	2	5.70	38.00	13.38	1.85	9.47	3.74	156	619	342.60
40	401	1793.40	4	22.90	108.80	55.99	3.63	16.55	9.02	79	783	318.70
44	401	1817.20	6	72.33	408.14	159.84	8.27	22.36	13.83	64	1598	545.00
48	401	1797.20	8	105.67	430.15	224.25	8.51	33.47	17.70	140	1727	687.20
52	801	3597.50	2	30.10	70.57	56.05	9.89	24.40	18.64	82	791	362.80
56	801	3589.20	4	94.42	491.09	189.64	16.24	79.86	33.32	66	832	517.30
60	801	3571.90	6	164.83	803.24	460.76	20.61	91.63	49.34	265	1509	605.50
64	801	3608.80	8	402.48	1564.07	719.01	31.57	107.47	58.93	310	3194	851.30
68	1201	5415.70	2	47.09	146.78	83.11	16.56	50.89	31.11	148	1050	471.60
72	1201	5384.90	4	168.73	614.03	364.31	32.15	103.32	70.07	212	686	402.00
76	1201	5398.90	6	514.27	2618.41	1192.11	66.12	270.39	138.81	420	2117	993.50
80	1201	5393.00	8	1136.37	3071.61	1779.11	112.17	288.15	185.88	280	3802	1171.20

https://doi.org/10.1002/net.21815

TABLE A23 NetMaker instances, highly robust efficiency, $k = 2$, random scenarios

	$\|V\|$	$\|E\|$	r	EL-Highly min	EL-Highly max	EL-Highly avg	RL-Highly min	RL-Highly max	RL-Highly avg	sols min	sols max	sols avg
3	101	444.95	2	0.00	0.03	0.01	0.00	0.01	0.00	0	2	1.10
7	101	448.35	4	0.02	0.06	0.03	0.00	0.01	0.01	0	1	0.55
11	101	449.35	6	0.04	0.14	0.08	0.01	0.02	0.01	0	1	0.25
15	101	447.60	8	0.07	0.21	0.11	0.01	0.02	0.01	0	1	0.25
19	201	908.60	2	0.02	0.05	0.03	0.01	0.02	0.01	0	3	1.35
23	201	897.05	4	0.04	0.16	0.09	0.01	0.03	0.02	0	1	0.40
27	201	901.05	6	0.10	0.33	0.20	0.02	0.04	0.03	0	1	0.25
31	201	887.20	8	0.21	0.55	0.35	0.03	0.05	0.04	0	1	0.25
35	401	1798.25	2	0.04	0.15	0.08	0.01	0.05	0.03	0	2	0.75
39	401	1798.15	4	0.19	0.43	0.29	0.04	0.08	0.05	0	1	0.45
43	401	1790.20	6	0.40	2.12	0.73	0.05	0.16	0.08	0	1	0.20
47	401	1795.75	8	0.65	2.86	1.55	0.06	0.18	0.12	0	1	0.35
51	801	3617.85	2	0.13	0.47	0.22	0.04	0.13	0.07	0	2	0.80
55	801	3621.45	4	0.66	1.31	0.96	0.10	0.18	0.14	0	2	0.45
59	801	3605.35	6	1.30	3.63	2.44	0.14	0.28	0.22	0	1	0.35
63	801	3618.15	8	2.64	7.24	5.06	0.18	0.38	0.30	0	1	0.30
67	1201	5393.00	2	0.21	0.59	0.43	0.06	0.16	0.12	0	3	0.75
71	1201	5406.85	4	1.25	2.61	1.93	0.19	0.33	0.25	0	2	0.25
75	1201	5394.40	6	2.94	9.12	5.02	0.25	0.60	0.39	0	1	0.20
79	1201	5408.90	8	5.52	16.18	9.29	0.38	0.93	0.56	0	1	0.10
83	1601	7188.15	2	0.43	1.01	0.65	0.12	0.25	0.17	0	3	1.25
87	1601	7240.00	4	1.88	3.67	2.60	0.29	0.61	0.41	0	1	0.35
91	1601	7196.85	6	4.22	10.60	7.15	0.39	0.77	0.57	0	1	0.15
95	1601	7184.00	8	7.22	20.34	13.95	0.49	1.14	0.81	0	1	0.15

TABLE A24 NetMaker instances, highly robust efficiency, $k = 3$, random scenarios

	$\|V\|$	$\|E\|$	r	EL-Highly min	EL-Highly max	EL-Highly avg	RL-Highly min	RL-Highly max	RL-Highly avg	sols min	sols max	sols avg
4	101	453.40	2	0.19	1.53	0.75	0.07	0.52	0.28	12	71	35.90
8	101	449.40	4	0.50	19.02	4.19	0.14	2.77	0.74	6	25	12.80
12	101	437.10	6	1.43	7.75	3.26	0.26	1.00	0.51	4	13	7.90
16	101	453.90	8	4.93	62.67	16.76	0.69	4.16	1.54	3	15	8.30
20	201	900.30	2	0.54	6.69	3.02	0.22	1.54	0.87	19	94	44.90
24	201	896.10	4	3.89	21.42	10.25	0.83	3.80	1.86	5	40	16.80
28	201	896.50	6	7.23	48.70	25.10	1.15	4.90	2.61	2	30	10.20
32	201	903.70	8	13.03	142.01	66.90	1.16	8.94	4.86	3	12	6.50
36	401	1805.20	2	5.58	41.46	13.69	1.85	9.56	3.74	20	107	53.10
40	401	1793.40	4	23.04	101.53	57.26	3.67	16.09	8.83	4	28	16.90
44	401	1817.20	6	77.99	432.07	170.98	8.89	22.13	13.68	1	29	12.70
48	401	1797.20	8	104.31	449.94	233.68	8.56	30.01	17.05	0	23	7.60
52	801	3597.50	2	30.75	77.63	63.80	9.93	24.13	18.69	25	141	61.80
56	801	3589.20	4	84.78	419.81	174.53	16.97	80.77	33.61	4	41	21.10
60	801	3571.90	6	166.80	782.04	429.02	20.46	90.55	48.08	3	23	10.90
64	801	3608.80	8	351.56	1467.07	659.68	31.51	107.18	58.12	4	19	8.40
68	1201	5415.70	2	50.67	149.00	85.00	16.40	48.45	30.19	17	168	60.10
72	1201	5384.90	4	174.05	617.25	365.62	32.65	102.50	69.82	3	58	17.40
76	1201	5398.90	6	493.57	2448.50	1218.99	64.34	270.00	135.88	5	25	15.00
80	1201	5393.00	8	1216.22	3391.55	1895.47	103.63	267.30	175.94	2	17	9.90

https://doi.org/10.1002/net.21815

A.3. Min-Ordering and Max-Ordering Scalarization Methods for Multi-Objective Robust Optimization

submitted to the *European Journal of Operational Research* in January 2018 and available as preprint (Schmidt et al., 2018)

Authors: Marie Schmidt, Anita Schöbel, Lisa Thom

Min-ordering and max-ordering scalarization methods for multi-objective robust optimization

Marie Schmidt[a], Anita Schöbel[b], and Lisa Thom[b,*]

[a]Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University Rotterdam, postal address: PO Box 1738, 3000 DR Rotterdam, The Netherlands, email address: schmidt2@rsm.nl
[b]Institut für Numerische und Angewandte Mathematik, Universität Göttingen, postal address: Lotzestr. 16-18, 37083 Göttingen, Germany, email addresses: schoebel@math.uni-goettingen.de (Anita Schöbel), l.thom@math.uni-goettingen.de (Lisa Thom)
[*]Corresponding author, email address: l.thom@math.uni-goettingen.de

Abstract

Several robustness concepts for multi-objective uncertain optimization have been developed during the last years, but not many solution methods. In this paper we introduce two methods to find minmax robust efficient solutions based on scalarizations: the min-ordering and the max-ordering method. We show that all point-based minmax robust weakly efficient solutions can be found with the max-ordering method and that the min-ordering method finds set-based minmax robust weakly efficient solutions, some of which cannot be found with formerly developed scalarization based methods. We then show how the scalarized problems may be approached for multi-objective uncertain combinatorial optimization problems with special uncertainty sets. We develop compact mixed-integer linear programming formulations for multi-objective extensions of bounded uncertainty. For interval uncertainty, we show that the resulting problems reduce to well-known single-objective problems.

1 Introduction

When applying optimization techniques to real-world problems, one often encounters the difficulties, that several objectives need to be optimized at the same time and that not all parameters are known exactly in advance. In multi-objective optimization several objectives are optimized simultaneously by choosing a *(Pareto) efficient* solution that cannot be improved in one objective without worsening it in another objective. Robust optimization is a way to handle uncertainties, without having to assume any information on probability distributions, hedging against (all) possible outcomes. During the last years, concepts of those fields have been combined to *multi-objective robust optimization*.

Several concepts on how to define robust solutions in multi-objective optimization have been developed. The common (single-objective) concept of *minmax robustness* aims to find a solution that minimizes the objective function in the worst case. One generalization to multi-objective optimization, which we call *point-based minmax robust efficiency*, was first introduced by [KL12]. They consider the worst case in each objective independently, which results in a deterministic multi-objective problem with bottleneck objective functions, called the robust counterpart. However, the resulting worst case point for a solution can differ significantly from the possible outcomes. Therefore, a second generalization of minmax robustness for multiple objectives has been developed by [EIS14]. They look at the outcome set of a solution under every scenario and compare these sets to each other to find so-called *set-based minmax robust efficient* solutions. A comparison of these two and other concepts for robust efficiency can be found in [IS16] and [WD16].

Common methods to find efficient solutions in the deterministic case, i.e. without uncertainty, are so-called *scalarization methods*, where the multi-objective problem is transformed to a family of single-objective problems, whose solutions are (weakly) efficient for the original problem. By solving the resulting problems, several different (and possibly all) efficient solutions are found. For an overview on scalarization methods see, e.g., [Ehr06].

In the uncertain case, several methods to find minmax robust efficient solutions have been developed, which are based on scalarizations: on the weighted sum and ϵ-constraint scalarization ([EIS14]), on the augmented weighted Chebyshev scalarization ([Ide14]) and on p-norm scalarizations ([BF17]). Point-based minmax robust efficient solutions can also be found by applying deterministic scalarization methods to the robust counterpart (see, e.g., [HNS13, KL12, FW14]).

In this paper we introduce two new methods to find minmax robust efficient solutions based on scalarizations: the *max-ordering* and *min-ordering* method, resulting in problems of the form min-max-max respective min-max-min. The min-ordering problem can therefore be interpreted as a so-called *adjustable robust problem* [BTGGN04], where only part of the decisions has to be made before the realization of the uncertain parameters.

In robust optimization, the considered uncertainty set, i.e., the possible values the uncertain parameters can attain, plays an important role w.r.t. solvability and complexity of the resulting robust problems. In this paper we investigate the min-ordering and max-ordering optimization problems for multi-objective minmax robust combinatorial optimization problems with specific uncertainty sets: One popular assumption is that each parameter attains a value in a given interval independently of the realization of the other parameters (*interval uncertainty*). Based on this, [BS03] introduced the (single-objective) concept of *bounded uncertainty*, assuming that the parameters vary in intervals, but the worst case is not attained for all parameters simultaneously. Uncertainty sets for multi-objective optimization based on bounded uncertainty have been considered in [DKW12, WLD$^+$17] (only considering uncertainty in the constraints) and [HNS13, RSST18] (resulting in an objective-wise uncertainty set). We introduce an extension of bounded uncertainty to multi-objective optimization for the case that the uncertainties in the objectives are not independent of each other.

Solution approaches for multi-objective minmax robust combinatorial problems with objective-wise bounded uncertainty have been developed in [RSST18]. [KRSS16] consider bi-objective robust combinatorial problems with finite and polyhedral uncertainty sets for several robustness concepts. The multi-objective robust version of the shortest path problem with finite uncertainty set is considered in [RSST17], where labeling algorithms are extended in order to find robust efficient solutions.

This paper is structured as follows: First, we give a short introduction to multi-objective robust optimization. In Section 3 we introduce the min-ordering and max-ordering optimization problems and show their general properties. In Section 4 we consider combinatorial multi-objective optimization problems with particular uncertainty sets and investigate the complexity and solvability of the resulting min-ordering and max-ordering problems.

2 Preliminaries

In this section we introduce some general notation and give a short introduction to multi-objective optimization and multi-objective robust optimization.
Throughout this paper, we use the symbols $<$ (strictly less than) and $\leqq$ (less than or equal to) to compare values in $\mathbb{R}$. Further, ∂M denotes the boundary of a set $M \subseteq \mathbb{R}^k$ and we use $i \in [k]$ as an abbreviation for $i \in \{1, ..., k\}$.
To shorten the text we use a [. /.] notation, e.g., instead of "x is smaller than y if $x < y$ and x is smaller than or equal to y if $x \leqq y$" we write "x is smaller than [$\cdot$/or equal to] y if $x[< / \leqq]y$".

2.1 Multi-objective robust optimization

Definition 1. *Given a set $\mathcal{X}$ of feasible solutions and $k \in \mathbb{N}$ objective functions $z_1, ..., z_k : \mathcal{X} \to \mathbb{R}$, we call*

$$\min_{x \in \mathcal{X}} z(x) = \begin{pmatrix} z_1(x) \\ \vdots \\ z_k(x) \end{pmatrix}$$

a multi-objective optimization problem (MOP).

If $k = 1$ we say that the problem is a *single-objective* problem. For $k \geqq 2$, a solution that minimizes all objectives simultaneously does usually not exist. Therefore, we use the concept of *efficient solutions.*

Definition 2. *For two vectors $y^1, y^2 \in \mathbb{R}^k$ we use the notation*

$$\begin{aligned} y^1 \leq y^2 &\Leftrightarrow y_i^1 < y_i^2 \text{ for } i \in [k], \\ y^1 \leq y^2 &\Leftrightarrow y_i^1 \leqq y_i^2 \text{ for } i \in [k] \text{ and } y^1 \neq y^2, \\ y^1 \leqq y^2 &\Leftrightarrow y_i^1 \leqq y_i^2 \text{ for } i \in [k]. \end{aligned}$$

We also define $\mathbb{R}^k_{[>/\geq/\geqq]} := \{y \in \mathbb{R}^k : 0[< / \leq / \leqq]y\}$.

Definition 3. *A solution $x \in \mathcal{X}$ is a* [weakly/$\cdot$/strictly] efficient solution *for (MOP), if there is no $x' \in \mathcal{X}$ such that $z(x')[< / \leq / \leqq]z(x)$.*

Note that a solution $x \in \mathcal{X}$ is [weakly/$\cdot$/strictly] efficient if and only if there is no $x' \in X$ with

$$z(x') \in z(x) - (\mathbb{R}^k_{[>/\geq/\geqq]}).$$

We now assume that the input data is uncertain, i.e., not all parameters are exactly known in advance. Instead, they depend on a scenario, which will only be revealed after one has chosen a solution. The set $\mathcal{U}$ of all possible scenarios is called the *uncertainty set.*

Definition 4. *Given a feasible set of solutions $\mathcal{X}$, an uncertainty set $\mathcal{U}$, and a multi-objective function $z : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^k$, the family of multi-objective optimization problems*

$$\left(\min_{x \in \mathcal{X}} z(x, \xi), \xi \in \mathcal{U} \right)$$

is called a multi-objective uncertain optimization problem *(MOUP).*

In the following we assume $\mathcal{X}$ and $\mathcal{U}$ to be compact and non-empty and the z_i to be continuous in x and ξ. If a problem or part of a problem is not subject to uncertainty, we say that it is *deterministic*, e.g., this is the case for a (MOUP) with $|\mathcal{U}| = 1$.
Note that the formulation in Definition 4 only considers uncertainty in the objective function. If the constraints, i.e., the set of feasible solutions, are subject to uncertainty, we aim to find solutions which are feasible in all scenarios (as proposed in the seminal works on robustness, see, e.g., [Soy73, BTN98]). For this purpose, the sets of feasible solutions under all scenarios can be intersected in advance to obtain a (deterministic) set of *robust feasible solutions*. Hence, in the following, we assume the feasible set $\mathcal{X}$ to be deterministic.
To decide what is a good solution for a multi-objective uncertain problem is not trivial. In single-objective robust optimization one looks for so-called robust optimal solutions. Often these are defined as solutions, which have a minimal worst case value, i.e., one solves $\min_{x\in\mathcal{X}} \max_{\xi\in\mathcal{U}} z(x,\xi)$ (see, e.g., [BTEGN09]). This concept has been generalized to robust efficiency for multi-objective problems in various ways (e.g., [KL12, EIS14]), since the notion of *worst case* is not clear in the multi-objective case.
We present the two most common concepts for minmax robust efficiency: *point-based minmax robust efficiency* and *set-based minmax robust efficiency*. For point-based minmax robust efficiency, we determine the worst case for each solution x and objective i individually, and compare the solutions w.r.t. the resulting point $\bar{z}(x)$. For set-based minmax robust efficiency, we check whether there exists a solution $\xi \in \mathcal{U}$ with $\{z(x,\xi) : \xi \in \mathcal{U}\} \subseteq \{z(x,\xi) : \xi \in \mathcal{U}\} - \mathbb{R}^k_{\geq}$ (analogous to determining efficiency in the deterministic case by checking whether a solution $x' \in \mathcal{X}$ with $z(x') \in z(x) - \mathbb{R}^k_{\geq}$ exists).

Definition 5 ([KL12], [EIS14])**.** *Given a multi-objective uncertain optimization problem, we define*

$$\bar{z}(x) := \begin{pmatrix} \max_{\xi\in\mathcal{U}} z_1(x,\xi) \\ \vdots \\ \max_{\xi\in\mathcal{U}} z_k(x,\xi) \end{pmatrix}$$

A solution $x \in \mathcal{X}$ is point-based minmax robust [weakly/·/strictly] efficient *for (MOUP) (abbreviated: pointMR [weakly/·/strictly] efficient), if it is a [weakly/·/strictly] efficient solution for the* robust counterpart $\min_{x\in\mathcal{X}} \bar{z}(x)$, *i.e., if there is no $x' \in \mathcal{X}$ with*

$$\bar{z}(x') \in \bar{z}(x) - \mathbb{R}^k_{[>/\geq/\geqq]}.$$

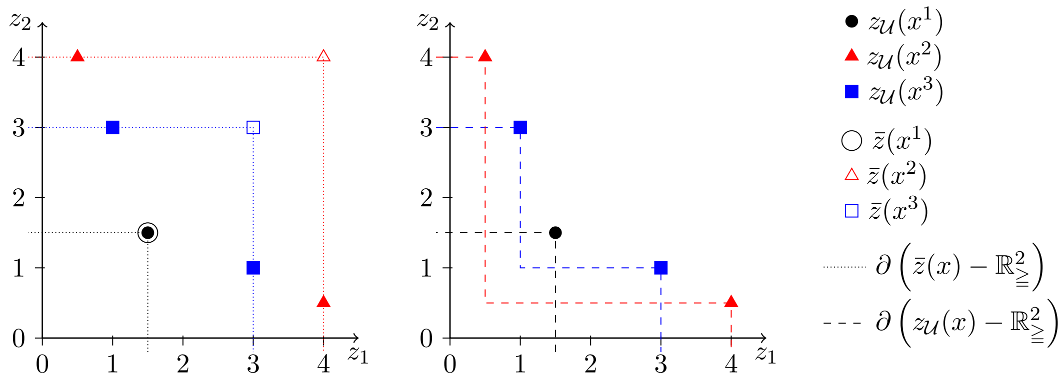
Defining

$$z_{\mathcal{U}}(x) := \{z(x,\xi) : \xi \in \mathcal{U}\},$$

a solution $x \in \mathcal{X}$ is set-based minmax robust [weakly/·/strictly] efficient *for (MOUP) (abbreviated: setMR [weakly/·/strictly] efficient), if there exists no $x' \in \mathcal{X}$ with*

$$z_{\mathcal{U}}(x') \subseteq z_{\mathcal{U}}(x) - \mathbb{R}^k_{[>/\geq/\geqq]}.$$

Both concepts reduce to minmax robustness for $k = 1$, i.e., the pointMR efficient solutions and setMR efficient solutions are then identical to the solutions of $\min_{x\in\mathcal{X}} \max_{\xi\in\mathcal{U}} z_1(x,\xi)$. Note that every pointMR [weakly/strictly] efficient solution is also setMR [weakly/strictly] efficient and that the two concepts coincide, if (MOUP) is *objective-wise uncertain*, i.e., if $\mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_k$ and $z_i(x,\xi) = z_i(x,\xi_i), \xi_i \in \mathcal{U}_i \;\forall\; i \in [k]$.



(a) Only x^1 is pointMR efficient. (b) All solutions are setMR efficient.

Figure 1: Determining pointMR efficient solutions and setMR efficient solutions for the instance in Example 6.

Example 6. *Let a multi-objective uncertain optimization problem be given with* $\mathcal{X} := \{x^1, x^2, x^3\}$, $\mathcal{U} := \{\xi^1, \xi^2\}$ *and*

$$z(x^1, \xi^1) = z(x^1, \xi^2) = (1.5, 1.5)$$
$$z(x^2, \xi^1) = (0.5, 4), z(x^2, \xi^2) = (4, 0.5)$$
$$z(x^3, \xi^1) = (1, 3), z(x^3, \xi^2) = (3, 1).$$

Figure 1(a) shows $\bar{z}(x)$ *and* $\partial(\bar{z}(x) - \mathbb{R}^k_{\geq})$ *and Figure 1(b) shows* $z_{\mathcal{U}}(x)$ *and* $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geq})$ *for* $x \in \mathcal{X}$. *All three solutions are setMR efficient, whereas only* x^1 *is pointMR efficient.*

The following lemma characterizes setMR efficient solutions.

Lemma 7 ([EIS14])**.** *Given a multi-objective uncertain optimization problem (MOUP). For all* $x, x' \in \mathcal{X}$,

$$z_{\mathcal{U}}(x') \subseteq z_{\mathcal{U}}(x) - \mathbb{R}^k_{[>/\geq/\geqq]} \Leftrightarrow \forall\ \xi \in \mathcal{U}\ \exists\ \eta \in \mathcal{U} : z(x', \xi)[< / \leq / \leqq]z(x, \eta).$$

2.2 Methods to find robust efficient solutions based on scalarizations

In (deterministic) multi-objective optimization it is common to find a set of efficient solutions with a *scalarization method*, i.e., by solving a family of single-objective, so-called *scalarized*, problems (see, e.g., [Ehr06]). For finding pointMR efficient solutions, these methods can directly be applied to the robust counterpart $\min_{x \in \mathcal{X}} \bar{z}(x)$. In case of set-based minmax robust efficiency, the extension of scalarization methods is not as straightforward, because the robust counterpart is a set-valued problem. The following methods to find setMR efficient solutions based on scalarizations have been developed.
[EIS14] introduce two methods based on scalarizations: The *weighted sum scalarization method* and the *ϵ-constraint method*, which are extensions of the corresponding methods for the deterministic case. They show that both methods find setMR weakly efficient solutions. The solutions for the weighted sum scalarized problems are even setMR efficient, if

the weights are chosen strictly greater than zero. The solutions found with the *ϵ-constraint method* are always pointMR weakly efficient. The authors show that the two methods do not always find the same solutions and that there can exist setMR efficient solutions, which cannot be found by either of these methods.
[Ide14] introduce a method based on the (augmented) weighted Chebyshev scalarization with reference point 0. [Ide14] show that all solutions found with this *(augmented) weighted Chebyshev method* are setMR weakly efficient. In case of objective-wise uncertainty, the scalarized problem in [Ide14] is identical to the scalarized problem in [HNS13] (if the robust utopian point in [HNS13] can be chosen as 0), where the deterministic augmented weighted Chebychev method is applied to the robust counterpart $\min_{x\in\mathcal{X}} \bar{z}(x)$ to find pointMR efficient solutions.
[BF17] consider order-preserving scalarizing functions $s : \mathbb{R}^k \to \mathbb{R}$ and the resulting scalarized problems $\min_{x\in\mathcal{X}} \max_{\xi\in\mathcal{U}} s(z(x))$. They show that for so-called *strongly increasing* scalarizing functions the solutions for the scalarized problem are setMR efficient. In an application they consider weighted p-norms as scalarizing functions, resulting in the *p-norm scalarization method* (e.g., the weighted sum scalarization method for $p = 1$).

3 Min-ordering and max-ordering method for multi-objective uncertain problems

Definition 8. *Let*

$$(P) \quad \left(\min_{x\in\mathcal{X}} z(x,\xi), \xi \in \mathcal{U} \right)$$

be a multi-objective uncertain optimization problem. For a given weight vector $\lambda \in \mathbb{R}^k_>$ and reference point $r \in \mathbb{R}^k$ we define the corresponding min-ordering optimization problem *as*

$$(P\text{-}min(r,\lambda)) \quad \min_{x\in\mathcal{X}} \max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i \left(z_i(x,\xi) - r_i\right)$$

and the corresponding max-ordering optimization problem *as*

$$(P\text{-}max(r,\lambda)) \quad \min_{x\in\mathcal{X}} \max_{\xi\in\mathcal{U}} \max_{i\in[k]} \lambda_i \left(z_i(x,\xi) - r_i\right).$$

We further denote the objective value for a given $x \in \mathcal{X}$ by

$$\alpha^{min}(x,r,\lambda) := \max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i(z_i(x,\xi) - r_i) \qquad \text{for } (P\text{-}min(r,\lambda)),$$
$$\alpha^{max}(x,r,\lambda) := \max_{\xi\in\mathcal{U}} \max_{i\in[k]} \lambda_i(z_i(x,\xi) - r_i) \qquad \text{for } (P\text{-}max(r,\lambda)).$$

Note that $\alpha^{\min}(x,r,\lambda)$ and $\alpha^{\max}(x,r,\lambda)$ exist for all $x \in \mathcal{X}$ because $\mathcal{U}$ is compact and nonempty and the finitely many functions $z_i(x,\cdot) : \mathcal{U} \to \mathbb{R}$ are continuous. The values $\alpha^{\min}(x,r,\lambda)$ and $\alpha^{\max}(x,r,\lambda)$ also have a geometric interpretation, which we detail in Section 3.1.
In Sections 3.2 and 3.3, we show that optimal solutions for (P-min(r,λ)) and (P-max(r,λ)) are setMR weakly efficient and solutions for (P-max(r,λ)) even pointMR weakly efficient. Similar to the existing methods discussed in Section 2.2, we obtain a *min-ordering* resp. *max-ordering scalarization method* to find a set of robust efficient solutions by varying the

parameters r, λ and solving the resulting problems (P-min(r, λ)) resp. (P-max(r, λ)). The max-ordering scalarization method is similar to the weighted Chebyshev method for multi-objective robust problems given in [Ide14] (and for objective-wise uncertainty in [HNS13]), but with arbitrary reference point.

Before investigating properties of the solutions for (P-min(r, λ)) and (P-max(r, λ)), we provide a brief example to give an intuition on their meaning for the original problem:
Consider a student organization who wants to offer cheap lunch for students in several university towns and has to decide on a dish $x \in \mathcal{X}$ in advance. They can price the dish differently in each town and because of a very small profit margin the price depends on the prices of the ingredients in the supermarket in town. They aim to minimize the lunch prices in all towns simultaneously, i.e., $z_i(x, \xi)$ is the price of dish x in town i, where the uncertainty in the price development is modeled by $\xi \in \mathcal{U}$. Solving (P-max(r, λ)) with $r = (0, ..., 0)^T, \lambda = (1, ..., 1)^T$ means then to minimize the highest price any student in any town has to pay for their meal in the worst case. Solving (P-min(r, λ)) with the same r, λ means to minimize the best price the organization can offer in some university, assuming the worst price development. I.e., this is the price p they can legitimately use in their advertisement "Cheap student lunch - starting from p!", because in some town the price will not be higher than p.

The remainder of this section is structured as follows: We first give a geometric interpretation of the problems (P-min(r, λ)) and (P-max(r, λ)) and a characterization of their solutions in Section 3.1. We then investigate properties of the solutions found with the max-ordering method in Section 3.2 and with the min-ordering method in Section 3.3.
In Section 4 we show how (P-min(r, λ)) and (P-max(r, λ)) can be solved for multi-objective uncertain combinatorial problems with particular uncertainty sets and investigate their complexity. For this, we use the following reformulations of (P-min(r, λ)) and (P-max(r, λ)) in case of a single scenario.

Remark 9. *If the uncertainty set $\mathcal{U}$ contains only one scenario ξ, i.e., (MOUP) is a deterministic problem, (P-min(r, λ)) then reduces to $\min_{x \in \mathcal{X}, i \in [k]} \lambda_i(z_i(x, \xi) - r_i)$. This can be solved by solving the k single-objective deterministic problems*

$$(P_i) \quad \min_{x \in \mathcal{X}} \lambda_i(z_i(x, \xi) - r_i)$$

and choosing the best of the obtained solutions.
(P-max(r, λ)) reduces to $\min_{x \in \mathcal{X}} \max_{i \in [k]} \lambda_i(z_i(x, \xi) - r_i)$, which can be interpreted as a single-objective minmax robust problem with a discrete uncertainty set.

3.1 Geometric interpretation of (P-max(r, λ)) and (P-min(r, λ))

The sublevel set of the function $\max_{i \in [k]} \lambda_i(z_i - r_i)$ for level $\alpha \in \mathbb{R}$ is

$$\begin{aligned} L_{\leqq}^{\max, r, \lambda}(\alpha) &= \left\{ z \in \mathbb{R}^k : \max_{i \in [k]} \lambda_i(z_i - r_i) \leqq \alpha \right\} \\ &= \left\{ z \in \mathbb{R}^k : z_i \leqq \frac{\alpha}{\lambda_i} + r_i \ \forall i \ \in [k] \right\} \\ &= \left\{ z \in \mathbb{R}^k : z \leqq \alpha \left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k} \right)^T + r \right\}. \end{aligned}$$

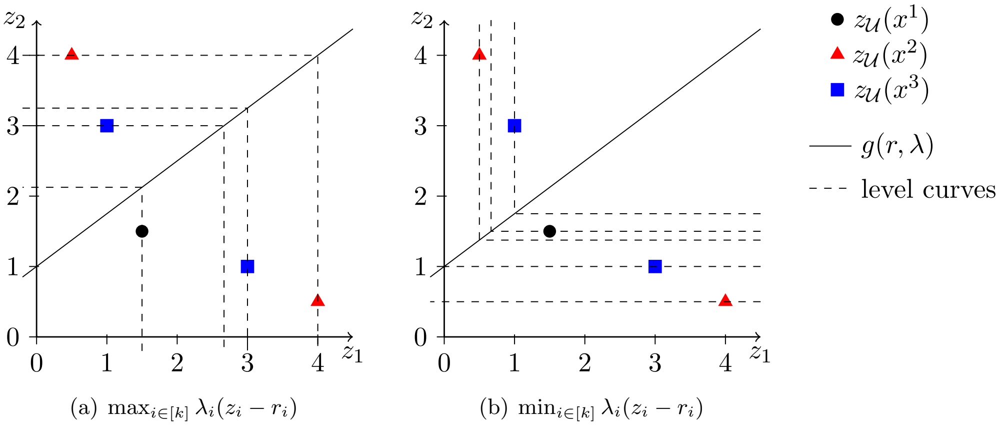


Figure 2: Level curves of the functions $\max_{i\in[k]} \lambda_i(z_i - r_i)$ and $\min_{i\in[k]} \lambda_i(z_i - r_i)$ with $r = (0,1)^T, \lambda = (3,4)^T$, which contain any $z(x,\xi)$ from Example 6.

and that of the function $\min_{i\in[k]} \lambda_i(z_i - r_i)$ is

$$\begin{aligned} L_{\leqq}^{\min,r,\lambda}(\alpha) &= \left\{ z \in \mathbb{R}^k : \min_{i\in[k]} \lambda_i(z_i - r_i) \leqq \alpha \right\} \\ &= \left\{ z \in \mathbb{R}^k : \exists i \in [k] \text{ with } z_i \leqq \frac{\alpha}{\lambda_i} + r_i \right\} \\ &= \left\{ z \in \mathbb{R}^k : z \not> \alpha \left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k} \right)^T + r \right\}. \end{aligned}$$

Therefore, every sublevel set of $\max_{i\in[k]} \lambda_i(z_i - r_i)$ or $\min_{i\in[k]} \lambda_i(z_i - r_i)$ can be uniquely identified with a point on the line

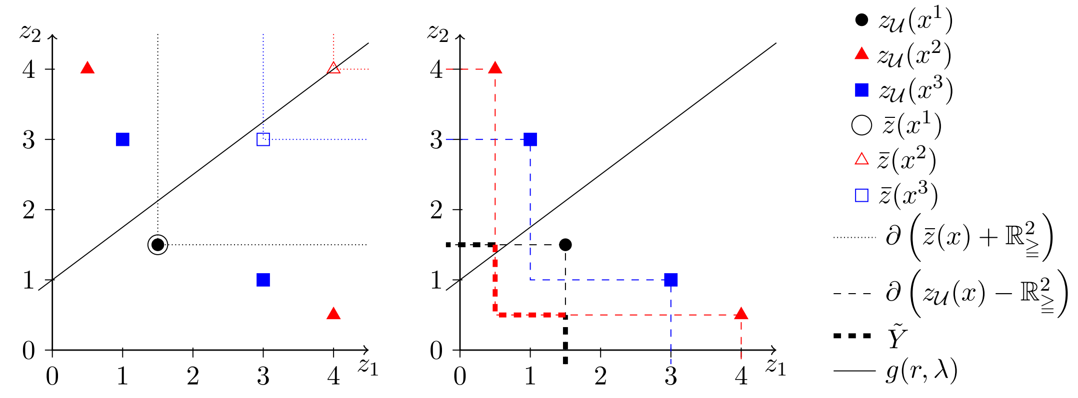
$$g(r,\lambda) := \left\{ y(\alpha) := r + \alpha \left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k} \right)^T : \alpha \in \mathbb{R} \right\}.$$

For two points $y(\alpha), y(\alpha') \in g(r,\lambda)$ we have $y(\alpha) \leq y(\alpha') \Leftrightarrow y(\alpha) < y(\alpha') \Leftrightarrow \alpha < \alpha'$, because of $\lambda_i > 0 \; \forall i \in [k]$. Figure 2 shows the level curves of $\max_{i\in[k]} \lambda_i(z_i - r_i)$ and $\min_{i\in[k]} \lambda_i(z_i - r_i)$ for $r = (0,0)^T$ and $\lambda = (2,1)^T$ that contain $z(x,\xi)$ for some $x \in \mathcal{X}$ and $\xi \in \mathcal{U}$ from Example 6.

Recall the definitions of $\bar{z}(x)$ and $z_{\mathcal{U}}(x)$, used in the definition of pointMR efficiency and setMR efficiency (Definition 5). The following theorem shows that the optimal solutions for (P-max(r,λ)) can be identified by comparing the intersection points of $g(r,\lambda)$ with $\partial(\bar{z}(x) + \mathbb{R}^k_{\geqq})$ for all $x \in \mathcal{X}$. Similarly, the optimal solutions of (P-min(r,λ)) can be identified by comparing the intersection points of $g(r,\lambda)$ with $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$ for all $x \in \mathcal{X}$.

Theorem 10. *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_{>}$ be given. A feasible solution $x^* \in \mathcal{X}$ is optimal for (P-max(r,λ)) if and only if there exists $y^* \in \mathbb{R}^k$ such that (x^*, y^*) is an efficient solution for*

$$(G\text{-}max(r,\lambda)) \qquad \begin{aligned} \min \;& y \\ s.t. \; y \;&\in\; g(r,\lambda) \cap \partial(\bar{z}(x) + \mathbb{R}^k_{\geqq}) \\ x \;&\in\; \mathcal{X}. \end{aligned}$$



(a) Finding solutions for (P-max(r,λ)) (b) Finding solutions for (P-min(r,λ))

Figure 3: Determining the intersection point of $g(r,\lambda)$ with $\partial(\bar{z}(x)+\mathbb{R}^2_{\geqq})$ (a) and $\partial(z_{\mathcal{U}}(x)-\mathbb{R}^2_{\geqq})$ (b) for the solutions in Example 6. As an example, $g(r,\lambda)$ is shown for $r=(0,1)^T, \lambda=(3,4)^T$.

A feasible solution $x^* \in \mathcal{X}$ is optimal for (P-min(r,λ)) if and only if there exists $y^* \in \mathbb{R}^k$ such that (x^*, y^*) is an efficient solution for

$$
\begin{array}{llrcl}
(\textit{G-min}(r,\lambda)) & & \min & y & \\
& & s.t. & y & \in & g(r,\lambda)\cap\partial(z_{\mathcal{U}}(x)-\mathbb{R}^k_{\geqq}) \\
& & & x & \in & \mathcal{X}.
\end{array}
$$

Proof. We first show

$$g(r,\lambda)\cap\partial(\bar{z}(x)+\mathbb{R}^k_{\geqq}) = \left\{ r+\alpha^{max}(x,r,\lambda)\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \right\}$$

for every $x\in\mathcal{X}, r\in\mathbb{R}^k, \lambda\in\mathbb{R}^k_{>}$. For every $\alpha\in\mathbb{R}$ with $\alpha>\alpha^{\max}(x,r,\lambda)$ we have

$$
\begin{aligned}
& \alpha > \alpha^{\max}(x,r,\lambda) = \max_{\xi\in\mathcal{U}}\max_{i\in[k]}\lambda_i(z_i(x,\xi)-r_i) \geqq \lambda_i(\max_{\xi\in\mathcal{U}} z_i(x,\xi)-r_i) && \forall\, i\in[k] \\
\Rightarrow\ & r_i+\alpha\cdot\frac{1}{\lambda_i} > \max_{\xi\in\mathcal{U}} z_i(x,\xi) = \bar{z}_i(x) && \forall\, i\in[k] \\
\Rightarrow\ & r+\alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \in \bar{z}(x)+\mathbb{R}^k_{>} = (\bar{z}(x)+\mathbb{R}^k_{\geqq})\setminus\partial(\bar{z}(x)+\mathbb{R}^k_{\geqq}).
\end{aligned}
$$

Further, for every $\alpha\in\mathbb{R}$ with $\alpha<\alpha^{\max}(x,r,\lambda)$,

$$
\begin{aligned}
& \alpha < \alpha^{\max}(x,r,\lambda) = \max_{\xi\in\mathcal{U}}\max_{i\in[k]}\lambda_i(z_i(x,\xi)-r_i) \\
\Rightarrow\ & r_i+\alpha\cdot\frac{1}{\lambda_i} < \max_{\xi\in\mathcal{U}} z_i(x,\xi) = \bar{z}_i(x) && \text{for at least one } i\in[k] \\
\Rightarrow\ & r+\alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \notin \bar{z}(x)+\mathbb{R}^k_{\geqq} \\
\Rightarrow\ & r+\alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \notin \partial(\bar{z}(x)+\mathbb{R}^k_{\geqq}) && \text{since } \bar{z}(x)+\mathbb{R}^k_{\geqq} \text{ is closed.}
\end{aligned}
$$

It follows that $y(\alpha^{max}(x,r,\lambda))$ is the unique intersection point of $g(r,\lambda)$ with $\partial(\bar{z}(x)+\mathbb{R}^k_{\geqq})$. Hence, the only $y \in \mathbb{R}^k$, such that (x,y) is feasible for (G-max(r,λ)), is $y(\alpha^{max}(x,r,\lambda))$. It follows that

$$\begin{aligned}
&x^* \text{ is optimal for(P-max}(r,\lambda))\\
\Leftrightarrow\ &\nexists\ x \in \mathcal{X} : \alpha^{max}(x,r,\lambda) < \alpha^{max}(x^*,r,\lambda)\\
\Leftrightarrow\ &\nexists\ x \in \mathcal{X} : y(\alpha^{max}(x,r,\lambda)) \leq y(\alpha^{max}(x^*,r,\lambda))\\
\Leftrightarrow\ &(x^*, y(\alpha^{max}(x^*,r,\lambda))) \text{ is an efficient solution for (G-max}(r,\lambda)).
\end{aligned}$$

Similarly, we show

$$g(r,\lambda) \cap \partial(\bar{z}(x)+\mathbb{R}^k_{\geqq}) = \left\{ r + \alpha^{max}(x,r,\lambda)\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \right\}$$

for every $x \in \mathcal{X}, r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_{>}$. For every $\alpha \in \mathbb{R}$ with $\alpha > \alpha^{\min}(x,r,\lambda)$ we have

$$\begin{aligned}
&\alpha > \alpha^{\min}(x,r,\lambda) \geqq \min_{i\in[k]} \lambda_i(z_i(x,\xi)-r_i) && \forall\ \xi \in \mathcal{U}\\
\Rightarrow\ &\forall\ \xi \in \mathcal{U}\ \exists\ i \in [k] : r_i + \alpha \cdot \frac{1}{\lambda_i} > z_i(x,\xi)\\
\Rightarrow\ &r + \alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \notin z(x,\xi) - \mathbb{R}^k_{\geqq} && \forall \xi \in \mathcal{U}\\
\Rightarrow\ &r + \alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \notin z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}\\
\Rightarrow\ &r + \alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \notin \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}), && \text{since } z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq} \text{ is closed,}
\end{aligned}$$

and for every $\alpha \in \mathbb{R}$ with $\alpha < \alpha^{\min}(x,r,\lambda)$,

$$\begin{aligned}
&\alpha < \alpha^{\min}(x,r,\lambda) = \min_{i\in[k]} \lambda_i(z_i(x,\xi)-r_i) && \text{for at least one } \xi \in \mathcal{U}\\
\Rightarrow\ &\exists \xi \in \mathcal{U} \text{ such that } \forall i \in [k] : r_i + \alpha \cdot \frac{1}{\lambda_i} < z_i(x,\xi)\\
\Rightarrow\ &\exists \xi \in \mathcal{U} : r + \alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \in z(x,\xi) - \mathbb{R}^k_{>}\\
\Rightarrow\ &r + \alpha\left(\frac{1}{\lambda_1},\ldots,\frac{1}{\lambda_k}\right)^T \in z_{\mathcal{U}}(x) - \mathbb{R}^k_{>} = (z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}) \setminus \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}).
\end{aligned}$$

Hence, for all $x \in \mathcal{X}$, $y(\alpha^{max}(x,r,\lambda))$ is the unique intersection point of $g(r,\lambda)$ with $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$. Therefore, x^* is optimal for (P-max(r,λ)) if and only if $(x^*, y(\alpha^{max}(x^*,r,\lambda)))$ is an efficient solution for (G-max(r,λ)). □

Note that it follows from the proof of Theorem 10 that for (G-max(r,λ)) and (G-min(r,λ)) every weakly efficient solution is also efficient, because we have $y(\alpha) \leq y(\alpha') \Leftrightarrow y(\alpha) < y(\alpha')$ for two points $y(\alpha), y(\alpha') \in g(r,\lambda)$. Theorem 10 implies, that a solution $x \in \mathcal{X}$ can be found with the [max-ordering/min-ordering] method if and only if there exist $\lambda \in \mathbb{R}^k_{>}, r \in \mathbb{R}^k, y \in \mathbb{R}^k$, such that (x,y) is (weakly) efficient for [(G-max(r,λ))/(G-min(r,λ))].

Figure 3 illustrates $g(r, \lambda)$, $\partial(\bar{z}(x) + \mathbb{R}^k_{\geqq})$ and $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$ for the feasible solutions in Example 6. It is easy to see in Figure 3(a) that for each choice of r, λ the intersection point of $g(r, \lambda)$ with $\partial(\bar{z}(x^1) + \mathbb{R}^k_{\geqq})$ has smaller coordinates than the intersection point of $g(r, \lambda)$ with $\partial(\bar{z}(x^2) + \mathbb{R}^k_{\geqq})$ or $\partial(\bar{z}(x^3) + \mathbb{R}^k_{\geqq})$, hence x^1 is the unique optimal solution for (P-max(r, λ)).
Let us now consider the sets

$$Y := \bigcup_{x \in \mathcal{X}} \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq}) \text{ and } \tilde{Y} := \{y \in Y : \nexists y' \in Y : y' < y\}.$$

For each $y \in \tilde{Y}$ there exists $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_{>}, x \in \mathcal{X}$ such that (x, y) is efficient for (G-min(r, λ)): choose $r = y$, then $y \in g(r, \lambda)$, hence there exists x such that (x, y) is feasible for (G-min(r, λ)), because $y \in Y$. Further there is no feasible (x', y') with $y' < y$, because $y \in \tilde{Y}$, hence (x, y) is (weakly) efficient for (G-min(r, λ)).
Figure 3(b) shows $\partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_{\geqq})$ for all $x \in \mathcal{X}$ in Example 6 as dashed lines and $\tilde{Y}$ as thick dashed line. Since $\partial(z_{\mathcal{U}}(x^1) - \mathbb{R}^k_{\geqq}) \cap \tilde{Y}$ and $\partial(z_{\mathcal{U}}(x^2) - \mathbb{R}^k_{\geqq}) \cap \tilde{Y}$ are not empty, x^1 and x^2 can be found with the min-ordering method. On the other hand, it is easy to see that for every $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_{>}$ there exists a point $\tilde{y} \in \tilde{Y} \cap g(r, \lambda)$ and that $\tilde{y} \leq y$ for $y \in g(r, \lambda) \cap \partial(z_{\mathcal{U}}(x^3) - \mathbb{R}^k_{\geqq})$. Therefore, x^3 is not optimal for (P-min(r, λ)).

3.2 Solutions found with the max-ordering method

[Ide14] show that (for fixed reference point 0) every [$\cdot$/unique] solution of (P-max(r, λ)) is setMR [weakly/strictly] efficient. We show that for every reference point $r \in \mathbb{R}^k$ every [$\cdot$/unique] solution of the max-ordering optimization problem is even pointMR [weakly/strictly] efficient and that for a small enough r all pointMR weakly efficient solutions can be found by choosing an appropriate λ.

Theorem 11. *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_{>}$ be given and let x be an optimal solution for (P-max(r, λ)). Then*

1. *x is pointMR weakly efficient for (P) and*
2. *if x is the unique optimal solution for (P-max(r, λ)), then x is pointMR strictly efficient.*

Proof. Let x be [an/the unique] optimal solution for (P-max(r, λ)). Assume that x is not pointMR [weakly/strictly] efficient. Then there exists a solution $x' \in \mathcal{X}$ with

$$\begin{aligned}
& \max_{\xi \in \mathcal{U}} z_i(x', \xi) \ [< / \leqq] \max_{\xi \in \mathcal{U}} z_i(x, \xi) && \forall i \in [k] \\
\Leftrightarrow \quad & \max_{\xi \in \mathcal{U}} \lambda_i \left(z_i(x', \xi) - r_i\right) \ [< / \leqq] \max_{\xi \in \mathcal{U}} \lambda_i \left(z_i(x, \xi) - r_i\right) && \forall i \in [k] \\
\Rightarrow \quad & \max_{i \in [k]} \max_{\xi \in \mathcal{U}} \lambda_i \left(z_i(x', \xi) - r_i\right) \ [< / \leqq] \max_{i \in [k]} \max_{\xi \in \mathcal{U}} \lambda_i \left(z_i(x, \xi) - r_i\right) \\
\Rightarrow \quad & \max_{\xi \in \mathcal{U}} \max_{i \in [k]} \lambda_i \left(z_i(x', \xi) - r_i\right) \ [< / \leqq] \max_{\xi \in \mathcal{U}} \max_{i \in [k]} \lambda_i \left(z_i(x, \xi) - r_i\right)
\end{aligned}$$

1. If x is not pointMR weakly efficient, i.e., $<$ holds, this is a contradiction to x being an optimal solution for (P-max(r, λ)).
2. If x is not pointMR strictly efficient, i.e., $\leqq$ holds, then x is not optimal for (P-max(r, λ)) or x' is optimal as well. This contradicts x being the unique optimal solution.

□

Theorem 11 implies that not all setMR weakly efficient solutions can be found with the max-ordering method, because a setMR weakly efficient solution is not necessarily pointMR weakly efficient. However, the following theorem shows that for a suitable choice of r all pointMR weakly efficient solutions can be found by varying λ.

Theorem 12. *Let x be a pointMR weakly efficient solution and let a reference point $r \in \mathbb{R}^k$ with $r_i < \max_{\xi\in\mathcal{U}} z_i(x,\xi)$ $\forall i \in [k]$ be given. Then there exists a weight vector $\lambda \in \mathbb{R}^k_>$ such that x is an optimal solution for (P-max(r,λ)).*

Proof. Because of $r_i < \max_{\xi\in\mathcal{U}} z_i(x,\xi)$ we obtain well-defined and positive weights by setting

$$\lambda_i := \frac{1}{\max_{\xi\in\mathcal{U}} z_i(x,\xi) - r_i} \;\forall\; i \;= 1,...,k.$$

It follows that $\max_{\xi\in\mathcal{U}} \lambda_i \left(z_i(x,\xi) - r_i\right) = \lambda_i \left(\max_{\xi\in\mathcal{U}} z_i(x,\xi) - r_i\right) = 1 \;\forall\; i \in [k]$.
Let $x' \in X$ be any feasible solution. Since x is weakly pointMR efficient, there exists at least one index $j \in \{1,...,k\}$ with $\max_{\xi\in\mathcal{U}} z_j(x,\xi) \leqq \max_{\xi\in\mathcal{U}} z_j(x',\xi)$. It follows that

$$\begin{aligned}\max_{i\in[k]}\max_{\xi\in\mathcal{U}} \lambda_i \left(z_i(x,\xi) - r_i\right) = 1 = \max_{\xi\in\mathcal{U}} \lambda_j \left(z_j(x,\xi) - r_j\right) &= \lambda_j \left(\max_{\xi\in\mathcal{U}} z_j(x,\xi) - r_j\right) \\ &\leqq \lambda_j \left(\max_{\xi\in\mathcal{U}} z_j(x',\xi) - r_j\right) = \max_{\xi\in\mathcal{U}} \lambda_j \left(z_j(x',\xi) - r_j\right) \\ &\leqq \max_{i\in[k]}\max_{\xi\in\mathcal{U}} \lambda_i \left(z_i(x',\xi) - r_i\right),\end{aligned}$$

hence, x is optimal for $(Pmax(r,\lambda))$. □

The results from Section 3.1 provide a geometric interpretation of the proof of Theorem 12: For given r, x and the λ constructed in the proof of Theorem 12, $g(r,\lambda)$ is the line through r and $\bar{z}(x)$. Then, $\bar{z}(x) = y(\alpha^{max}(x,r,\lambda))$ and $y(\alpha) \in \bar{z}(x) - \mathbb{R}^k_{\geq}$ for all $\alpha < (\alpha^{max}(x,r,\lambda))$. If x is pointMR efficient, $\bar{z}(x) - \mathbb{R}^k_{\geq} \cap \partial(\bar{z}(x') + \mathbb{R}^k_{\geq})$ is empty for all $x' \in \mathcal{X}$, hence $(x, y(\alpha^{max}(x,r,\lambda))$ is an efficient solution for (G-max(r,λ)).

3.3 Solutions found with the min-ordering method

For (P-min(r,λ)) we show that every [·/unique] solution is set-based robust [weakly/strictly] efficient, i.e., the min-ordering scalarization method is suitable for finding setMR (weakly) efficient efficient solutions. Moreover, we show that with this method we can find setMR efficient solutions that cannot be found with the other known scalarization methods presented in Section 2.2, including the weighted sum, ϵ-constraint and augmented weighted Chebychev method. This also implies that solutions for (P-min(r,λ)) are not necessarily pointMR efficient. In addition, we briefly discuss the connection to adjustable robustness.

Theorem 13. *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ be given and let x be an optimal solution for (P-min(r,λ)). Then*

1. *x is setMR weakly efficient for (P) and*
2. *if x is the unique optimal solution for (P-min(r,λ)), then x is setMR strictly efficient.*

Proof. Let x be [an/the unique] optimal solution for (P-min(r,λ)). Assume that x is not setMR [weakly/strictly] efficient. From Lemma 7 it follows that there exists a feasible solution x' with $\forall\ \xi \in \mathcal{U}\ \exists\ \eta \in \mathcal{U} : z(x',\xi)[< / \leqq]z(x,\eta)$. Let $\xi' \in \operatorname{argmax}_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i(z_i(x',\xi) - r_i)$ be a worst case scenario of x' w.r.t. (P-min(r,λ)) Then there exists $\eta' \in \mathcal{U}$ with

$$\begin{aligned} & z_i(x',\xi')[< / \leqq]z_i(x,\eta') && \forall\ i \in [k] \\ \Leftrightarrow\quad & \lambda_i\left(z_i(x',\xi') - r_i\right)[< / \leqq]\lambda_i\left(z_i(x,\eta') - r_i\right) && \forall\ i \in [k] \end{aligned}$$

We hence conclude that

$$\begin{aligned} \max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i\left(z_i(x',\xi) - r_i\right) \quad &= \quad \min_{i\in[k]} \lambda_i\left(z_i(x',\xi') - r_i\right) \\ &[< / \leqq] \quad \min_{i\in[k]} \lambda_i\left(z_i(x,\eta') - r_i\right) \\ &\leqq \quad \max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i\left(z_i(x,\xi) - r_i\right). \end{aligned}$$

1. If x is not setMR weakly efficient, i.e., $<$ holds, this is a contradiction to x being an optimal solution for (P-min(r,λ)).
2. If x is not setMR strictly efficient, i.e., $\leqq$ holds, then x is not optimal for (P-min(r,λ)) or x' is optimal as well. This contradicts x being the unique optimal solution of (P-min(r,λ)).

□

Theorem 14. *There exists a multi-objective uncertain optimization problem with setMR efficient solutions that cannot be found*

- *with the ϵ-constraint method,*
- *with the p-norm scalarization method,*
- *or by solving any scalarized problem of the form*

$$\min_{x\in\mathcal{X}} \left(\rho_1 \max_{\xi\in\mathcal{U}} \max_{i\in[k]} \nu_i\left(z_i(x,\xi) - r_i\right) + \rho_2 \max_{\xi\in\mathcal{U}} \sum_{i\in[k]} \mu_i\left(z_i(x,\xi) - r_i\right) \right) \tag{1}$$

  with $\rho \in \mathbb{R}^2_{\geq}, \nu, \mu \in \mathbb{R}^k_{>}, r_i \in \mathbb{R}^k$.

Some, but not all, of these solutions can be found with the min-ordering optimization method.

Proof. Consider the multi-objective uncertain optimization problem given in Example 6. Recall that all three solutions are setMR efficient. Because of

$$\begin{aligned} &\left(\max_{\xi\in\mathcal{U}} z_1(x^1,\xi), \max_{\xi\in\mathcal{U}} z_2(x^1,\xi)\right) = (1.5, 1.5) \\ < &\left(\max_{\xi\in\mathcal{U}} z_1(x^3,\xi), \max_{\xi\in\mathcal{U}} z_2(x^3,\xi)\right) = (3, 3) \\ < &\left(\max_{\xi\in\mathcal{U}} z_1(x^2,\xi), \max_{\xi\in\mathcal{U}} z_2(x^2,\xi)\right) = (4, 4), \end{aligned}$$

x^1 is the only pointMR weakly efficient solution, hence only x^1 can be found with the ϵ-constraint method ([EIS14]).
[BF17] show that a solution $x \in \mathcal{X}$ can only be found with the p-norm scalarization method if

$$\nexists x' \in \mathcal{X} : z_{\mathcal{U}}(x') \in \text{Conv}(z_{\mathcal{U}}(x)) - \mathbb{R}^k_>,$$

where $\text{Conv}(z_{\mathcal{U}}(x))$ denotes the convex hull of $z_{\mathcal{U}}(x)$. Since $(1.5, 1.5) \in \text{Conv}(\{(1,3),(3,1)\}) - \mathbb{R}^k_>$ and $(1.5, 1.5) \in \text{Conv}(\{(0.5,4),(4,0.5)\}) - \mathbb{R}^k_>$, x^1 is the only solution that can be found with the p-norm scalarization method.
Let now $\rho \in \mathbb{R}^2_\geq, \nu, \mu \in \mathbb{R}^k_>, r_i \in \mathbb{R}^k$ be given and consider the scalarized problem (1). We define

$$f(x) := \max_{\xi \in \mathcal{U}} \max_{i \in [k]} \nu_i \left(z_i(x,\xi) - r_i\right) \text{ and}$$
$$h(x) := \max_{\xi \in \mathcal{U}} \sum_{i \in [k]} \mu_i \left(z_i(x,\xi) - r_i\right)$$

From Theorem 11 it follows that only x^1 can be optimal for $\min_{x\in\mathcal{X}} f(x)$, because it is the only pointMR weakly efficient solution. In the following we show that x^1 is also the only optimal solution for $\min_{x\in\mathcal{X}} h(x)$. Let $\mu \in \mathbb{R}^2_>, \mu_i \geqq \mu_j, \{i,j\} = \{1,2\}$. Then

$$\begin{aligned}
h(x^1) &= 1.5\mu_1 + 1.5\mu_2 - \mu_1 r_1 - \mu_2 r_2 && \leqq 3\mu_i - \mu_1 r_1 - \mu_2 r_2\\
h(x^3) &= \max\{3\mu_1 + \mu_2, \mu_1 + 3\mu_2\} - \mu_1 r_1 - \mu_2 r_2 && > 3\mu_i - \mu_1 r_1 - \mu_2 r_2\\
h(x^2) &= \max\{4\mu_1 + 0.5\mu_2, 0.5\mu_1 + 4\mu_2\} - \mu_1 r_1 - \mu_2 r_2 && > 3\mu_i - \mu_1 r_1 - \mu_2 r_2
\end{aligned}$$

It follows that x^1 is the unique optimal solution for $\min_{x\in\mathcal{X}} h(x)$. Since it is also uniquely optimal for $\min_{x\in\mathcal{X}} f(x)$, x^1 is the unique optimal solution for (1) for every $\rho \in \mathbb{R}^2_\geq$.
We conclude that the setMR efficient solutions x^2 and x^3 cannot be found with any of the methods listed in the statement.
In Figure 3(b) it is easy to see that there exists no $r \in \mathbb{R}^k$ and $\lambda \in \mathbb{R}^k_>$, such that the minimal intersection point of $g(r,\lambda)$ with $\bigcup_{x\in\mathcal{X}} \partial(z_{\mathcal{U}}(x) - \mathbb{R}^k_\geqq)$ is in $\partial(z_{\mathcal{U}}(x^3) - \mathbb{R}^k_\geqq)$. With Theorem 10 it follows that x^3 cannot be found with the min-ordering scalarization method either.
However, x^2 is optimal for (P-min(r,λ)) with $r = (0,0)^T, \lambda = (1,1)^T$, because

$$\max_{\xi\in\mathcal{U}} \min_{i=1,2} z_i(x^2,\xi) = 0.5 < \max_{\xi\in\mathcal{U}} \min_{i=1,2} z_i(x^3,\xi) = 1 < \max_{\xi\in\mathcal{U}} \min_{i=1,2} z_i(x^1,\xi) = 1.5.$$

□

Remark 15. *Note that (P-min(r,λ)) can be interpreted as a special case of a (single-objective)* two-stage *or* adjustable robust *problem, where $x \in \mathcal{X}$ must be chosen* here-and-now, *i.e., before the realization of the uncertain parameters, whereas the relevant objective may be chosen afterwards (*wait-and-see*). This can be modeled by introducing an additional variable y, which determines the choice of the objective. However, the additional variable changes the structure of the underlying problem P, e.g., if $\mathcal{X}$ is the feasible set of a particular combinatorial problem (e.g., the shortest path problem), the feasible set $\mathcal{X}\times\{1,...,k\}$ does not have the same structure (e.g., is not necessarily equivalent to the set of all paths in a graph). Also, the additional variable is integer, such that solution methods for robust adjustable counterparts of linear programs cannot be used to solve (P-min(r,λ)).*

4 Min-ordering and max-ordering optimization problem for multi-objective combinatorial problems

In Section 3 we have shown that all pointMR efficient solutions can be found with the max-ordering method, and the min-ordering method finds setMR efficient solutions, some of which are not found with any of the formerly developed scalarization based methods (see Section 2.2). On an example, we have shown the meaning of the particular solutions obtained with the min-ordering and the max-ordering method.
Now, we investigate how the problems (P-min(r,λ)) and (P-max(r,λ)) can be solved for combinatorial problems. We show that in case of interval uncertainty the uncertainty set can be reduced to one scenario, resulting in problems which have already been considered in the literature. For a multi-objective extension of the so-called *bounded uncertainty set* we develop compact mixed-integer linear programming (MILP) formulations, i.e., formulations without nested minimum and maximum functions.

We consider *multi-objective combinatorial problems with uncertain costs* (*MOUCO*): Let a finite set of elements $E = \{e_1, ..., e_n\}$ and a feasible set $\mathcal{X} \subseteq \{0,1\}^n$ be given. Each feasible solution $x \in \mathcal{X}$ represents a subset of E, which contains element e_j if and only if $x_j = 1$. Further, a cost matrix $c \in \mathbb{R}^{k\times n}$ is given, assigning a cost $c_{i,j}$ to element e_j in the i-th objective function for $i \in [k]$. The costs are uncertain, i.e., $c \in \mathcal{U} \subseteq \mathbb{R}^{k\times n}$. The k objective functions $z_i(x,c)$ hence depend on x and on the realization of the costs and are given as

$$z_i(x,c) := \sum_{j\in[n]} c_{i,j}x_j \ \forall x \in \mathcal{X}, c \in \mathcal{U}.$$

For a solution $x \in \mathcal{X}$ we write $|x| := \sum_{j\in[n]} x_j$.

4.1 Interval uncertainty

We use a straight-forward extension of the often used single-objective concept of interval uncertainty, where each uncertain parameter takes any value in a given interval, independent of the realization of the other parameters.

Definition 16. *Let lower bounds $\hat{c} \in \mathbb{R}^{k\times n}$ and interval lengths $\delta \in \mathbb{R}^{k\times n}_{\geqq}$ be given. We define the* interval uncertainty set

$$\mathcal{U}^I := \left\{c \in \mathbb{R}^{k\times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0,1] \ \forall i \in [k], j \in [n]\right\}.$$

The following theorem shows that in case of interval uncertainty it is sufficient to consider the upper bounds of the intervals, i.e., the uncertainty set can be reduced to a single scenario. Therefore, (P-min(r,λ)) can be solved by solving k single-objective deterministic combinatorial problems and (P-max(r,λ)) can be interpreted as a single-objective minmax robust combinatorial problem with discrete uncertainty set (see Remark 9).

Theorem 17. *Let (P) be a MOUCO with uncertainty set $\mathcal{U} := \mathcal{U}^I$. We define $\bar{c}_{i,j} := \hat{c}_{i,j} + \delta_{i,j}$ for all $j \in [n], i \in [k]$. Then (P-min(r,λ)) is equivalent to*

$$\min_{x\in\mathcal{X}, i\in[k]} \lambda_i \left(z_i(x,\bar{c}) - r_i\right)$$

and (P-max(r,λ)) is equivalent to

$$\min_{x\in\mathcal{X}}\max_{i\in[k]}\lambda_i\left(z_i(x,\bar{c})-r_i\right).$$

Proof. From $\bar{c}\in\mathcal{U}^I$ and $c_{i,j}\leqq\bar{c}_{i,j}\ \forall\ c\in\mathcal{U}^I, j\in[n], i\in[k]$ we conclude

$$\begin{aligned}
& & \lambda_i(z_i(x,c)-r_i) &\leqq \lambda_i(z_i(x,\bar{c})-r_i) & &\forall\ x\in\mathcal{X}, c\in\mathcal{U}^I, i\in[k]\\
&\Rightarrow & \min_{i\in[k]}\lambda_i(z_i(x,c)-r_i) &\leqq \min_{i\in[k]}\lambda_i(z_i(x,\bar{c})-r_i) & &\forall\ x\in\mathcal{X}, c\in\mathcal{U}^I\\
&\overset{\bar{c}\in\mathcal{U}^I}{\Rightarrow} & \max_{c\in\mathcal{U}^I}\min_{i\in[k]}\lambda_i(z_i(x,c)-r_i) &= \min_{i\in[k]}\lambda_i(z_i(x,\bar{c})-r_i) & &\forall\ x\in\mathcal{X}\\
&\Rightarrow & \min_{x\in\mathcal{X}}\max_{c\in\mathcal{U}^I}\min_{i\in[k]}\lambda_i(z_i(x,c)-r_i) &= \min_{x\in\mathcal{X}}\min_{i\in[k]}\lambda_i(z_i(x,\bar{c})-r_i), & &
\end{aligned}$$

where all minima and maxima exist due to the finiteness of $\mathcal{X}$, the compactness of $\mathcal{U}^I$ and the continuity of $z(x,\cdot):\mathcal{U}^I\to\mathbb{R}$. For (P-max$(r,\lambda)$) we analogously obtain

$$\min_{x\in\mathcal{X}}\max_{c\in\mathcal{U}^I}\max_{i\in[k]}\lambda_i(z_i(x,c)-r_i)=\min_{x\in\mathcal{X}}\max_{i\in[k]}\lambda_i(z_i(x,\bar{c})-r_i).$$

□

It follows that (P-min(r,λ)) with interval uncertainty set $\mathcal{U}^I$ is polynomially solvable if the single-objective deterministic problem can be solved in polynomial time. However, (P-max(r,λ)) is as complex as a single-objective minmax robust problem with discrete uncertainty set. This has been shown to be NP-hard for several combinatorial problems, which can be solved in polynomial time in the single-objective deterministic case, e.g., the shortest path, minimum spanning tree and assignment problem, see [KY97].

4.2 Bounded uncertainty

The concept of *bounded uncertainty*, also called Γ-uncertainty or cardinality constrained uncertainty, was introduced for single-objective optimization by [BS03]. Its idea is that it is unlikely that all uncertain parameters, which vary in intervals, attain their worst case value simultaneously. Therefore, the authors assume that not more than Γ parameters differ from their so-called *nominal* value.
We extend this idea to multi-objective uncertain combinatorial optimization by assuming, that at most a given number Γ of all cost parameters will deviate from their minimal value.

Definition 18. *Let $\hat{c}\in\mathbb{R}^{k\times n}, \delta\in\mathbb{R}^{k\times n}_{\geqq}$ and $\Gamma\in\mathbb{Z}$ with $0\leqq\Gamma\leqq(n\cdot k)$ be given. We define the* discretely bounded uncertainty set *as*

$$\mathcal{U}^d:=\left\{c\in\mathbb{R}^{k\times n}: c_{i,j}=\hat{c}_{i,j}+\beta_{i,j}\delta_{i,j}, \beta_{i,j}\in\{0,1\}\ \forall i\in[k], j\in[n], \sum_{i\in[k],j\in[n]}\beta_{i,j}\leqq\Gamma\right\}$$

[BS03] also allow more than Γ parameters to deviate from their minimal value if not all attain their maximal value, but deviate to a lesser extend. In the single-objective robust optimization case treated in [BS03], restricting to what extent the parameters may deviate in total leads to the same objective value as restricting the number of deviating parameters. However, Example 22 shows that this does not hold for (P-min(r,λ)). Therefore, we also consider the *continuously bounded uncertainty set*:

Definition 19. *Let* $\hat{c} \in \mathbb{R}^{k \times n}, \delta \in \mathbb{R}^{k \times n}_{\geqq}$ *and* $\Gamma \in \mathbb{Z}$ *with* $0 \leqq \Gamma \leqq (n \cdot k)$ *be given. We define the* continuously bounded uncertainty set

$$\mathcal{U}^c := \left\{ c \in \mathbb{R}^{k \times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0,1] \ \forall i \in [k], j \in [n], \sum_{i \in [k], j \in [n]} \beta_{i,j} \leqq \Gamma \right\}$$

If we can assume that the uncertainties in the objectives are independent of each other, another possibility to extend the idea of bounded uncertainty to multi-objective optimization is to restrict the deviation of the parameters for each objective independently. This has been done in [HNS13, RSST18]. They use an objective-wise extension of the concept of bounded uncertainty, which we will refer to as *objective-wise bounded uncertainty.*

Definition 20. *Let* $\hat{c} \in \mathbb{R}^{k \times n}, \delta \in \mathbb{R}^{k \times n}_{\geqq}$ *and* $\Gamma_i \in \mathbb{Z}$ *with* $0 \leqq \Gamma_i \leqq n \ \forall i \in [k]$ *be given. We define the* objective-wise bounded uncertainty set

$$\mathcal{U}^{owb} := \left\{ c \in \mathbb{R}^{k \times n} : c_{i,j} = \hat{c}_{i,j} + \beta_{i,j}\delta_{i,j}, \beta_{i,j} \in [0,1] \ \forall i \in [k], j \in [n], \sum_{j \in [n]} \beta_{i,j} \leqq \Gamma_i \ \forall \ i \in [k] \right\}.$$

In the following, we focus on discretely and continuously bounded uncertainty.

4.2.1 MILP-formulation for (P-max(r, λ)) with bounded uncertainty

In this section we introduce a MILP-formulation for (P-max(r, λ)) with discretely or continuously bounded uncertainty set. We show that we can apply the same approach that [HNS13] use to develop an augmented weighted Chebyshev method for multi-objective uncertain linear problems with objective-wise bounded uncertainty set.
In the following we show that for (P-max(r, λ)) we do not need to distinguish between the uncertainty sets $\mathcal{U}^d$ and $\mathcal{U}^c$. Moreover, even using $\mathcal{U}^{\text{owb}}$ results in an equivalent problem, if the bound Γ_i is the same for all objectives.

Lemma 21. *For given* $\mathcal{X} \subseteq \{0,1\}^n, \lambda \in \mathbb{R}^k_{>}, r \in \mathbb{R}^k, \hat{c}, \delta \in \mathbb{R}^{k \times n}, \Gamma = \Gamma_1 = \ldots = \Gamma_k \in \mathbb{Z}_{\geqq}$*:*

$$\begin{aligned} \min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}^d} \max_{i \in [k]} \lambda_i(z_i(x,c) - r_i) &= \min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}^c} \max_{i \in [k]} \lambda_i(z_i(x,c) - r_i) \\ &= \min_{x \in \mathcal{X}} \max_{\xi \in \mathcal{U}^{owb}} \max_{i \in [k]} \lambda_i(z_i(x,c) - r_i). \end{aligned}$$

Proof. Let $x \in \mathcal{X}, i \in [k]$ be given. Let $\pi : [n] \to [n]$ be a permutation such that $\delta_{i,\pi(1)}x_{\pi(1)} \geqq \delta_{i,\pi(2)}x_{\pi(2)} \geqq \ldots \geqq \delta_{i,\pi(n)}x_{\pi(n)}$. We construct the scenario c^* by setting $c^*_{i',j} := \hat{c}_{i',j} + \beta^*_{i',j}\delta_{i',j}$ for all $i' \in [k]$ and $j \in [n]$ with

$$\beta^*_{i',j} := \begin{cases} 1 & \text{for } i = i', j = \pi(l), 1 \leqq l \leqq \Gamma \\ 0 & \text{else.} \end{cases}$$

Then $\sum_{i' \in [k], j \in [n]} \beta^*_{i',j} = \Gamma$, hence $c^* \in \mathcal{U}^d$. Further, for any $\beta \in [0,1]^{k \times n}$ with $\sum_{j \in [n]} \beta_{i,j} \leqq \Gamma$ we have $\sum_{j \in [n]} \beta_{i,j}\delta_{i,j}x_j \leqq \sum_{j \in [n]} \beta^*_{i,j}\delta_{i,j}x_j$, because $\delta_{i,\pi(l)}x_{\pi(l),i} \leqq \delta_{i,\pi(l')}x_{i,\pi(l')}$ for $l \geqq l'$. Consequently,

$$z_i(x,c) = \sum_{j \in [n]} (\hat{c}_{i,j} + \beta_{i,j}\delta_{i,j})x_j \leqq \sum_{j \in [n]} (\hat{c}_{i,j} + \beta^*_{i,j}\delta_{i,j})x_j = z_i(x,c^*) \leqq \max_{c' \in \mathcal{U}^d} z_i(x,c') \ \forall \ c \in \mathcal{U}^{\text{owb}}$$

and therefore

$$\max_{c \in \mathcal{U}^{\text{owb}}} \lambda_i \left(z_i(x,c) - r_i\right) \leqq \max_{c \in \mathcal{U}^d} \lambda_i \left(z_i(x,c) - r_i\right).$$

Further,

$$\max_{c \in \mathcal{U}^d} \lambda_i \left(z_i(x,c) - r_i\right) \leqq \max_{c \in \mathcal{U}^c} \lambda_i \left(z_i(x,c) - r_i\right) \leqq \max_{c \in \mathcal{U}^{\text{owb}}} \lambda_i \left(z_i(x,c) - r_i\right),$$

because of $\mathcal{U}^d \subseteq \mathcal{U}^c \subseteq \mathcal{U}^{\text{owb}}$. Since these results hold for all $x \in \mathcal{X}, i \in [k]$, we get

$$\begin{aligned}
& \max_{c \in \mathcal{U}^d} \lambda_i \left(z_i(x,c) - r_i\right) = \max_{c \in \mathcal{U}^c} \lambda_i \left(z_i(x,c) - r_i\right) \\
& \qquad = \max_{c \in \mathcal{U}^{\text{owb}}} \lambda_i \left(z_i(x,c) - r_i\right) && \forall\, i \in [k], x \in \mathcal{X} \\
\Rightarrow\quad & \max_{i \in [k]} \max_{c \in \mathcal{U}^d} \lambda_i \left(z_i(x,c) - r_i\right) = \max_{i \in [k]} \max_{c \in \mathcal{U}^c} \lambda_i \left(z_i(x,c) - r_i\right) \\
& \qquad = \max_{i \in [k]} \max_{c \in \mathcal{U}^{\text{owb}}} \lambda_i \left(z_i(x,c) - r_i\right) && \forall\, x \in \mathcal{X} \\
\Rightarrow\quad & \min_{x \in \mathcal{X}} \max_{i \in [k]} \max_{c \in \mathcal{U}^d} \lambda_i \left(z_i(x,c) - r_i\right) = \min_{x \in \mathcal{X}} \max_{i \in [k]} \max_{c \in \mathcal{U}^c} \lambda_i \left(z_i(x,c) - r_i\right) \\
& \qquad = \min_{x \in \mathcal{X}} \max_{i \in [k]} \max_{c \in \mathcal{U}^{\text{owb}}} \lambda_i \left(z_i(x,c) - r_i\right),
\end{aligned}$$

where, again, all minima and maxima exist due to the finiteness of $\mathcal{X}$, the compactness of $\mathcal{U}^d$,$\mathcal{U}^c$,$\mathcal{U}^{\text{owb}}$ and the continuity of $z(x,\cdot) : [\mathcal{U}^d/\mathcal{U}^c/\mathcal{U}^{\text{owb}}] \to \mathbb{R}$. □

Because of this identity we can use the approach given in [HNS13] also for the uncertainty sets $\mathcal{U}^d$ or $\mathcal{U}^c$:

$$\begin{aligned}
\min_{x \in \mathcal{X}} \max_{c \in [\mathcal{U}^d/\mathcal{U}^c/\mathcal{U}^{\text{owb}}]} \max_{i \in [k]} \lambda_i (z_i(x,c) - r_i) &= \min_{x \in \mathcal{X}} \max_{c \in \mathcal{U}^{\text{owb}}} \max_{i \in [k]} \lambda_i (z_i(x,c) - r_i) \\
&= \min_{x \in \mathcal{X}} \max_{i \in [k]} \lambda_i \left(\max_{c \in \mathcal{U}^{\text{owb}}} z_i(x,c) - r_i \right)
\end{aligned}$$

is equivalent to

$$\begin{aligned}
\min\quad & y \\
\text{s.t.}\quad & y \geqq \lambda_i(\tilde{z}_i - r_i) && \forall\quad i \in [k] \\
& \tilde{z}_i \geqq \max\nolimits_{c \in \mathcal{U}^{\text{owb}}} z_i(x,c) && \forall\quad i \in [k] \\
& x \in \mathcal{X}.
\end{aligned}$$

As shown by [BS03], the dual of the single-objective problem $\max_{c \in \mathcal{U}^{\text{owb}}} z_i(x,c)$ is equivalent to the linear program

$$\begin{aligned}
\min\quad & \textstyle\sum_{j \in [n]} \hat{c}_{i,j} x_j + \theta\Gamma + \sum_{j \in [n]} \rho_j \\
\text{s.t.}\quad & \rho_j + \theta \geqq \delta_{i,j} x_j && \forall\, j \in [n] \\
& \theta \geqq 0 \\
& \rho_j \geqq 0 && \forall\, j \in [n].
\end{aligned}$$

Similar to [HNS13], we conclude that (P-max(r,λ)) is equivalent to

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & y \geqq \lambda_i(\tilde{z}_i - r_i) && \forall\, i \in [k] \\ & \tilde{z}_i - \textstyle\sum_{j\in[n]} \hat{c}_{i,j}x_j - \theta_i\Gamma - \textstyle\sum_{j\in[n]} \rho_{i,j} \geqq 0 && \forall\, i \in [k] \\ & \rho_{i,j} + \theta_i - \delta_{i,j}x_j \geqq 0 && \forall\, j \in [n], i \in [k] \\ & \rho_{i,j}, \theta_i \geqq 0 && \forall\, j \in [n], i \in [k] \\ & x \in \mathcal{X}. \end{aligned}$$

4.2.2 MILP-formulation for (P-min(r,λ)) with continuously bounded uncertainty

For a fixed x we can reformulate $\max_{c\in\mathcal{U}^c} \min_{i\in[k]} \lambda_i(z_i(x,c) - r_i)$ as following:

$$(M(x)) \quad \max_{c\in\mathcal{U}^c} \min_{i\in[k]} \lambda_i(z_i(x,c) - r_i)$$

$$\Leftrightarrow$$

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \leqq \lambda_i\Big(\sum_{j\in[n]} \hat{c}_{i,j}x_j + \sum_{j\in[n]} \beta_{i,j}\delta_{i,j}x_j - r_i\Big) && \forall\, i \in [k] \\ & \sum_{j\in[n], i\in[k]} \beta_{i,j} \leqq \Gamma \\ & \beta_{i,j} \in [0,1] && \forall\, j \in [n], i \in [k] \end{aligned}$$

Since $\beta_{i,j}$ only contributes to the objective function if $x_j \neq 0$ and 0 is the only lower bound on $\beta_{i,j}$, there is always an optimal solution with $x_j = 0 \Rightarrow \beta_{i,j} = 0\ \forall j \in [n], i \in [k]$. Hence, we can replace $\beta_{i,j}$ with $\tilde{\beta}_{i,j} := \beta_{i,j}x_j$. Further, $\tilde{\beta}_{i,j}x_j = \tilde{\beta}_{i,j}$, hence we obtain the equivalent problem

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \leqq \lambda_i\Big(\sum_{j\in[n]} \hat{c}_{i,j}x_j + \sum_{j\in[n]} \tilde{\beta}_{i,j}\delta_{i,j} - r_i\Big) && \forall\, i \in [k] \\ & \sum_{j\in[n], i\in[k]} \tilde{\beta}_{i,j} \leqq \Gamma \\ & \tilde{\beta}_{i,j} \leqq x_j && \forall\, j \in [n], i \in [k] \\ & \tilde{\beta}_{i,j} \geqq 0 && \forall\, j \in [n], i \in [k] \end{aligned}$$

and its dual

$$\begin{aligned} (D(x)) \quad \min \quad & \sum_{i\in[k], j\in[n]} \lambda_i\hat{c}_{i,j}x_j\tau_i - \lambda_i r_i\tau_i + \Gamma\pi + \sum_{j\in[n], i\in[k]} x_j\nu_{i,j} \\ \text{s.t.} \quad & \textstyle\sum_{i\in[k]} \tau_i = 1 \\ & -\lambda_i\delta_{i,j}\tau_i + \pi + \nu_{i,j} \geq 0 && \forall\, j \in [n], i \in [k] \\ & \tau_i, \pi, \nu_{i,j}, \geq 0 && \forall\, j \in [n], i \in [k] \end{aligned}$$

In order to use $(D(x))$ instead of $(M(x))$ as inner problem of (P-min(r, λ)), we replace $x_j \tau_i$ by the new variable $\tilde{\tau}_{i,j}$ and $x_j \nu_{i,j}$ by $\tilde{\nu}_{i,j}$. Since $x_j \in \{0, 1\}$, $\tau_i \geq 0$ and $\sum_{i \in [k]} \tau_i = 1 \Rightarrow \tau_i \leq 1$, we can ensure $\tilde{\tau}_{i,j} = x_j \tau_i$ by adding the constraints

$$\begin{aligned}
&\tilde{\tau}_{i,j} \leqq \tau_i \\
&\tilde{\tau}_{i,j} \leqq x_j \\
&\tilde{\tau}_{i,j} \geqq \tau_i - (1 - x_j) \\
&\tilde{\tau}_{i,j} \geqq 0.
\end{aligned}$$

Further, consider a feasible solution for $(D(x))$ with $\nu_{i,j} > \lambda_i \delta_{i,j}$. Since $\nu_{i,j}$ occurs in only one constraint, which requires

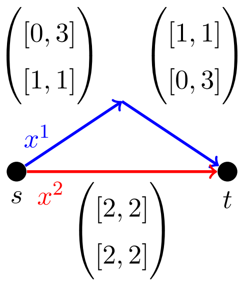
$$\nu_{i,j} \geqq \lambda_i \delta_{i,j} \tau_i - \pi,$$

we can choose $\nu_{i,j} = \lambda_i \delta_{i,j}$ instead and obtain a still feasible solution. Its objective value is not worse, since $\nu_{i,j}$ contributes with nonnegative factor to the objective function. Hence, we can restrict the feasible space of $(D(x))$ by adding the constraint $\nu_{i,j} \leqq \lambda_i \delta_{i,j}$. Then, the following constraints ensure that $\tilde{\nu}_{i,j} = x_j \nu_{i,j}$:

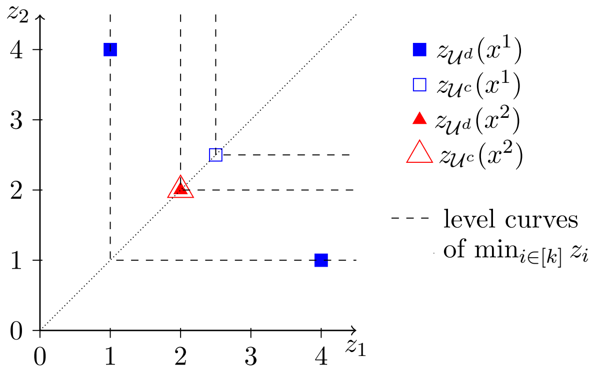
$$\begin{aligned}
&\tilde{\nu}_{i,j} \leqq \nu_{i,j} \\
&\tilde{\nu}_{i,j} \leqq x_j \lambda_i \delta_{i,j} \\
&\tilde{\nu}_{i,j} \geqq \nu_{i,j} - \lambda_i \delta_{i,j}(1 - x_j) \\
&\tilde{\nu}_{i,j} \geqq 0.
\end{aligned}$$

We obtain the following MILP-formulation for (P-min(r, λ)) with uncertainty set $\mathcal{U}^c$:

$$\begin{array}{llll}
\min & \sum\limits_{i \in [k], j \in [n]} \lambda_i \hat{c}_{i,j} \tilde{\tau}_i - \lambda_i r_i \tau_i + \Gamma \pi + \sum\limits_{j \in [n], i \in [k]} \tilde{\nu}_{i,j} & & \\
\text{s.t.} & \sum_{i \in [k]} \tau_i & = 1 & \\
& -\lambda_i \delta_{i,j} \tau_i + \pi + \nu_{i,j} & \geqq 0 & \forall\, j \in [n], i \in [k] \\
& \tilde{\tau}_{i,j} - \tau_i & \leqq 0 & \forall\, j \in [n], i \in [k] \\
& \tilde{\tau}_{i,j} - x_j & \leqq 0 & \forall\, j \in [n], i \in [k] \\
& \tilde{\tau}_{i,j} - \tau_i - x_j & \geqq -1 & \forall\, j \in [n], i \in [k] \\
& \tilde{\nu}_{i,j} - \nu_{i,j} & \leqq 0 & \forall\, j \in [n], i \in [k] \\
& \tilde{\nu}_{i,j} - x_j \lambda_i \delta_{i,j} & \leqq 0 & \forall\, j \in [n], i \in [k] \\
& \tilde{\nu}_{i,j} - \nu_{i,j} - \lambda_i \delta_{i,j} x_j & \geqq -\lambda_i \delta_{i,j} & \forall\, j \in [n], i \in [k] \\
& \tau_i, \tau_0, \nu_{i,j}, \tilde{\tau}_{i,j}, \tilde{\nu}_{i,j} & \geqq 0 & \forall\, j \in [n], i \in [k] \\
& x & \in \mathcal{X} &
\end{array}$$



(a) Graph in Example 22



(b) $z_{\mathcal{U}^d}(x)$, $z_{\mathcal{U}^c}(x)$ and level curves of $\min_{i\in[k]} z_i$

Figure 4: Example 22 shows that (P-min(r,λ)) with $\mathcal{U}^d$ is not equivalent to (P-min(r,λ)) with $\mathcal{U}^c$.

4.2.3 MILP formulation for (P-min(r,λ)) with discretely bounded uncertainty

In contrast to (P-max(r,λ)), the solutions for (P-min(r,λ)) with discretely bounded uncertainty can differ from the solution for (P-min(r,λ)) with continuously bounded uncertainty, as the following example shows.

Example 22. *Consider an instance of (P-min(r,λ)) with weights $\lambda = (1,1)^T$, reference point $r = (0,0)^T$, feasible set $\mathcal{X} = \{x^1 = (1,1,0), x^2 = (0,0,1)\}$ and discretely bounded uncertainty set $\mathcal{U}^d$ with $\Gamma = 1$. Our nominal costs are given by $\hat{c}$ and the interval lengths are given by δ as specified below:*

$$\hat{c} = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix}, \delta = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}.$$

The instance can for example be interpreted as an instance of the multi-objective robust shortest path problem in the graph shown in Figure 4(a).
Since only one cost value can deviate from its lower bound, we either have $z_1(x^1,c) = 0+1$ or $z_2(x^1,c) = 1+0$. Hence, $\max_{c\in\mathcal{U}^d} \min_{i\in[k]} z_i(x^1,c) = 1$.
However, if we consider the continuous bounded uncertainty set $\mathcal{U}^c$ with the same $\Gamma, \hat{c}, \delta$ instead of $\mathcal{U}^d$, by setting

$$\beta' = \begin{pmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & 0 \end{pmatrix},$$

we obtain the cost matrix

$$c' = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix} + \begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \end{pmatrix} = \begin{pmatrix} 1.5 & 1 & 2 \\ 1 & 1.5 & 2 \end{pmatrix}$$

Therefore, $\max_{c\in\mathcal{U}^c} \min_{i\in[k]} z_i(x^1,c) \geq 2.5$.
On the other hand we have $\max_{c\in\mathcal{U}^d} \min_{i\in[k]} z_i(x^2,c) = \max_{c\in\mathcal{U}^c} \min_{i\in[k]} z_i(x^2,c) = 2$. It follows that x^1 is the only optimal solution for (P-min(r,λ)) with uncertainty set $\mathcal{U}^d$, but x^2

is the only optimal solution for (P-min(r, λ)) with uncertainty set $\mathcal{U}^c$. The objective vectors $z(x, \xi)$ and the corresponding level curves are shown in Figure 4(b).

Therefore, the derived MILP-formulation for (P-min(r, λ)) with continuously bounded uncertainty is not valid for (P-min(r, λ)) with discretely bounded uncertainty. The example shows also, that the inner maximization problem of (P-min(r, λ)) is not equivalent to its linear relaxation. Hence, we cannot use the approach to dualize the linearly relaxed inner problem here. However, with help of the identity we prove in Theorem 27 we can nevertheless find a minimization problem which is equivalent to the inner maximization problem and derive a MILP formulation for (P-min(r, λ)) with discretely bounded uncertainty set.

Definition 23. *Let δ be a vector in $\mathbb{R}^n$ or a matrix in $\mathbb{R}^{k \times l}$ and let an index set $I \subseteq [n]$ resp. $I \subseteq [k] \times [l]$ be given. We denote the j-smallest of all entries δ_i with $i \in I$ as $j\text{-}\min_I \delta$ and the j-greatest as $j\text{-}\max_I \delta$. For $j = 0$ or $j > |I|$ we set $j\text{-}\min_I \delta = j\text{-}\max_I \delta = 0$.*

Notation 24. *For a binary vector $x \in \{0,1\}^n$ we write $I(x) := \{j \in [n] : x_j = 1\}$.*

Definition 25. *Let $r \in \mathbb{R}^k, \lambda \in \mathbb{R}^k_>$ and $x \in \mathcal{X}$ be given. We define $M \in \mathbb{R}^{k \times (\Gamma+1)}$ by its entries*

$$m_{i,l} := \lambda_i \left(-r_i + \sum_{j \in I(x)} \hat{c}_{i,j} + \sum_{h=1}^{l-1} h\text{-}\max_{I(x)} \delta_{(i,\cdot)} \right),$$

i.e., $\frac{m_{i,l}+r_i}{\lambda_i}$ is the sum of the nominal cost of x in the i-th objective and the l highest interval lengths $\delta_{i,j}$ among those with $x_j = 1$ w.r.t. the i-th objective.

Example 26. *Consider an instance of (P-min(r, λ)) with $r = (0,0,0)^T, \lambda = (1,3,1)^T$ and uncertainty set $\mathcal{U}^d$ with $\Gamma = 6$. Let a feasible solution x be given with $|x| = 6$ and*

$$\sum_{j \in I(x)} \hat{c}_{(\cdot,j)} = \begin{pmatrix} 10 \\ 4 \\ 14 \end{pmatrix}, \left\{ \delta_{(\cdot,j)} : j \in I(x) \right\} = \left\{ \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 5 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix}, \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} \right\}.$$

We exemplarily compute $m_{1,3}$. For the first objective, the highest interval length $\delta_{1,j}$ among those with $j \in I(x)$ is 5, the second highest 4 and the nominal cost 10. Hence, we obtain

$$m_{1,3} = \lambda_1 \left(-r_1 + \sum_{j \in I(x)} \hat{c}_{1,j} + \sum_{h=1}^{2} h\text{-}\max_{I(x)} \delta_{(1,\cdot)} \right) = 1(-0 + 10 + 5 + 4) = 19.$$

The complete matrix for this example is

$$M = \begin{pmatrix} 10 & 15 & 19 & 22 & 24 & 25 & 25 \\ 12 & 15 & 18 & 21 & 24 & 27 & 30 \\ 14 & 20 & 25 & 29 & 32 & 34 & 35 \end{pmatrix}.$$

Theorem 27. *Given $x \in \mathcal{X}$ and the corresponding matrix M, the optimal objective value z^* of the inner maximization problem of (P-min(r,λ)) equals the $(\Gamma+1)$-smallest entry in M, i.e.,*

$$z^* := \max_{c \in \mathcal{U}} \min_{i \in [k]} \lambda_i (z_i(x,c) - r_i) = (\Gamma+1)\text{-}\min_{[k]\times[\Gamma+1]} M =: m^*.$$

Proof. We show first, that $z^* \leqq m^*$. Let c^* with $c^*_{i,j} = \hat{c}_{i,j} + \beta^*_{i,j}\delta_{i,j}$ be an optimal solution of the inner maximization problem $\max_{c\in\mathcal{U}} \min_{i\in[k]} \lambda_i(z_i(x,c) - r_i)$ with objective value z^*. Let us now look at the structure of the cost matrix c^*, or, more precisely, at each row $c^*_{(i,\cdot)}$ of this matrix, representing the costs under objective i in scenario c^*. Let $l_i := \sum_{j\in[n]} \beta^*_{i,j}$ be the number of entries in this row which deviate from their nominal value. Since we maximize the costs we can w.l.o.g. assume that among all $i \in I(x)$ the l_i indices with highest entries in $\delta_{(i,\cdot)}$ are chosen to deviate from the nominal value.
Due to the construction of the matrix M, it follows that the objective value of x in scenario c^* with respect to objective i is equal to the (l_i+1)st entry of line $m_{i,\cdot}$:

$$\lambda_i(z_i(x,c^*) - r_i) = \lambda_i \left(-r_i + \sum_{j\in I(x)} \hat{c}_{i,j} + \sum_{h=1}^{l_i} h\text{-}\max_{I(x)} \delta_{(i,\cdot)} \right) = m_{i,(l_i+1)}.$$

M is constructed such that in each row i we have $m_{i,l} \leqq m_{i,l'} \ \forall\ l \leqq l'$. Hence, in row i there are at most l_i matrix entries smaller than m_{i,l_i+1} and in total there are at most $\sum_{i\in[k]} l_i = \sum_{i\in[k]}\sum_{j\in[n]} \beta^*_{i,j} \leqq \Gamma$ matrix entries smaller than $\min_{i\in[k]} m_{i,l_i+1}$. This implies

$$z^* = \min_{i\in[k]} m_{i,(l_i+1)} \leqq (\Gamma+1)\text{-}\min_{i\in[k], j\in[\Gamma+1]} M = m^*.$$

To show $z^* \geqq m^*$, we construct a scenario $\tilde{c} \in \mathcal{U}$ with objective value m^*. For each $i \in [k]$ we define

$$\hat{l}_i := \max\{l : m_{i,l} < m^*\}.$$

Because of $m_{i,l} \leqq m_{i,l'}\ \forall\, l \leqq l'$, we have $m_{i,l} < m^*\ \forall\, l \leqq \hat{l}_i$ and $m^* \leqq m_{i,(\hat{l}_i+1)} \leqq m_{i,l'}\ \forall\, l' > \hat{l}_i$ we conclude

$$\sum_{i=1}^{k} \hat{l}_i \leqq \Gamma \text{ and } m^* = \min_{i\in[k]} m_{i,(\hat{l}_i+1)}.$$

We construct a $\tilde{\beta}$ such that the solution $\tilde{c}$ with $\tilde{c}_{i,j} = \hat{c}_{i,j} + \tilde{\beta}_{i,j}\delta_{i,j}$ is feasible and has objective value m^*: For each $i \in [k]$ we choose a set $\hat{J}_i \subseteq I(x)$ of $\hat{l}_i$ indices with largest interval lengths, i.e., such that $|\hat{J}_i| = \hat{l}_i$ and $\delta_{i,j} \geqq \delta_{i,j'}\ \forall\ j \in \hat{J}_i, j' \in I(x) \setminus \hat{J}_i$. We set

$$\tilde{\beta}_{i,j} := \begin{cases} 1 & \text{for } j \in \hat{J}_i \\ 0 & \text{else} \end{cases} \qquad \text{and} \qquad \tilde{c}_{i,j} := \hat{c}_{i,j} + \tilde{\beta}_{i,j}\delta_{i,j}.$$

Then $\sum_{i\in[k],j\in[n]} \tilde{\beta}_{i,j} = \sum_{i\in[n]} \hat{l}_i \leqq \Gamma$, hence, $\tilde{c} \in \mathcal{U}^d$. Further,

$$\begin{aligned} z^* &\geqq \min_{i\in[k]} \lambda_i(z_i(x,\tilde{c}) - r_i) = \min_{i\in[k]} \lambda_i \left(\sum_{j\in[n]} (\hat{c}_{i,j}x_j + \tilde{\beta}_{i,j}\delta_{i,j}x_j) - r_i \right) \\ &= \min_{i\in[k]} \lambda_i \left(-r_i + \sum_{j\in[n]} \hat{c}_{i,j}x_j + \sum_{j\in\hat{J}_i} \delta_{i,j} \right) \\ &= \min_{i\in[k]} \lambda_i \left(-r_i + \sum_{j\in[n]} \hat{c}_{i,j}x_j + \sum_{h=0}^{\hat{l}_i} h\text{-}\max_{I(x)}\ \delta_{(i,\cdot)} \right) = \min_{i\in[n]} m_{i,(\hat{l}_i+1)} = m^*. \end{aligned}$$

□

Example 28. *Consider the instance in Example 26 and the feasible solution x. We have $\Gamma+1 = 7$ and the 7-th smallest entry in M is 19. It follows that* $\max_{c\in\mathcal{U}^d} \min_{i\in[k]} \lambda_i(z_i(x,c) - r_i) = 19$.

With help of this equality we derive a MILP formulation for (P-min(r,λ)). In a preprocessing step, for each $i \in [k]$ we sort the entries of the vector $\delta_{(i,\cdot)}$ decreasingly and set

$$y_{i,j,j'} := \begin{cases} 1 & \text{if } \delta_{i,j} \text{ before } \delta_{i,j'} \text{ w.r.t. this sorting} \\ 0 & \text{else} \end{cases}$$

Then, for a given x, we can formulate $\max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i(z_i(x,c) - r_i)$ as a minimization problem with the variables

z	being the objective value
$m_{i,l}$	representing $m_{i,l}$ as given in Definition 25
$w_{i,l}$	indicating if $m_{i,l}$ is one of the $\Gamma+1$ smallest entries of M
$u_{i,j,l}$	indicating if $\delta_{i,j}$ is one of the summands in $m_{i,l}$
q_l	indicating if x contains at least l elements

and the constants

$$N_i := \sum_{j\in[n]} (\hat{c}_{i,j} + \delta_{i,j})\ \forall i \in [k].$$

If x is known, many of the values can be precomputed. However, when using the problem as inner problem for (P-min(r,λ)), they are variables. We construct the following MILP formulation for $\max_{\xi\in\mathcal{U}} \min_{i\in[k]} \lambda_i(z_i(x,c)-r_i)$:

$$
\begin{aligned}
\min\quad & z && && \\
\text{s.t.}\quad & z \geqq m_{i,l} - (1-w_{i,l})N_i && \forall i\in[k], l\in[\Gamma+1] && (1)\\
& \sum_{\substack{i\in[k]\\ l\in[\Gamma+1]}} w_{i,l} = \Gamma+1 && && (2)\\
& m_{i,l} = \lambda_i\left(\sum_{j\in[n]} \hat{c}_{i,j}x_j + \sum_{j\in[n]} u_{i,j,l}\delta_{i,j} - r_i\right) && \forall i\in[k], l\in[\Gamma+1] && (3)\\
& \sum_{j\in[n]} u_{i,j,l} \geqq (l-1) - \Gamma q_l && \forall i\in[k], l\in[\Gamma+1] && (4)\\
& \sum_{j\in[n]} u_{i,j,l} \geqq \sum_{j\in[n]} x_j - |E|(1-q_l) && \forall i\in[k], l\in[\Gamma+1] && (5)\\
& u_{i,j,l} \leqq x_j && \forall j\in[n], i\in[k], l\in[\Gamma+1] && (6)\\
& u_{i,j',l} - u_{i,j,l} \leqq 1 - y_{i,j,j'}x_j && \forall j,j'\in[n], i\in[k], l\in[\Gamma+1] && (7)\\
& u_{i,j,l}, w_{i,l}, q_l \in \{0,1\} && \forall j\in[n], i\in[k], l\in[\Gamma+1] && (8)
\end{aligned}
$$

The first two constraints ensure that z, when minimized, is set to the $(\Gamma+1)$-smallest of the variables $m_{i,l}$. Because of Constraints (4) and (5), for each i and l at least $\min\{|x|, l-1\}$ of the $u_{i,j,l}$ are set to 1. Hence, at least $\min\{|x|, l-1\}$ of the $\delta_{i,j}$ are summed up in Constraint (3). Constraints (6) and (7) ensure, that these are the largest $\delta_{i,j}$ among those with $x_j = 1$. We obtain

$$\sum_{h=1}^{l-1} h\text{-}\max_{I(x)}\ \delta_{(i,\cdot)} \leq \sum_{j\in[n]} u_{i,j,l}\delta_{i,j}\ \forall\ l\in[\Gamma+1], i\in[k],$$

with equality in case of an optimal solution, because z is minimized, hence $m_{i,l}$ is minimized. Then, $m_{i,l}$ take exactly the values given in Definition 25 (Constraint (3)). We conclude that (P-min(r,λ)) with uncertainty set $\mathcal{U}^d$ can be formulated as

$$
\begin{aligned}
(\text{P-min}(r,\lambda))\quad \min\quad & z\\
\text{s.t.}\quad & (1)-(8)\\
& x\in\mathcal{X}.
\end{aligned}
$$

4.2.4 Complexity of (P-min(r,λ)) and (P-max(r,λ)) with bounded uncertainty

For $\Gamma = 0$, the uncertainty sets $\mathcal{U}^d$ and $\mathcal{U}^c$ only contain one scenario. From Remark 9 it hence follows, analogous to the case of interval uncertainty, that (P-min(r,λ)) is polynomially solvable, if the single-objective deterministic problem is polynomially solvable, whereas (P-max(r,λ)) is NP-hard for several combinatorial problems, e.g., the shortest path, minimum spanning tree and assignment problem.
The following Theorem shows that (P-min(r,λ)) with uncertainty set $\mathcal{U}^d$ is NP-hard for the shortest path and minimum spanning tree problem, if $\Gamma = 1$.

Theorem 29. *(P-min(r, λ)) with uncertainty set $\mathcal{U}^d$ and $\Gamma = 1$ is NP-hard for the shortest path problem and the minimum spanning tree problem, even for two objectives, $\lambda = (1,1)^T$ and $r = (0,0)^T$.*

Proof. We consider the single-objective minmax robust shortest path resp. minimum spanning tree problem with a discrete scenario set consisting of two scenarios. This has been proven to be NP-hard for both problems (see [KY97]). We reduce it to (P-min(r, λ)) with two objectives and discretely bounded uncertainty set with $\Gamma = 1$.
Let an instance I of the single-objective minmax robust problem be given. In case of the shortest path problem, we have given a graph G with edge set $E = \{e_1, \ldots, e_n\}$, and a start node s and end node t in G. The set of feasible solutions $\mathcal{X} \subseteq \{0,1\}^n$ contains all vectors that represent a simple path from s to t. In case of the minimum spanning tree problem, E is again the edge set of a graph G and the feasible solutions represent the spanning trees in G. Further, we have given two scenarios ξ^1, ξ^2 and edge costs $b \in \mathbb{R}^{2\times n}$, assigning cost $b_{i,j}$ to edge e_j under scenario ξ. We construct an instance I' of (P-min(r, λ)) as following:

- We start with the graph G from I and construct edge costs for the discretely bounded uncertainty set: $\hat{c}_{i,j} := b_{i,j}, \delta_{i,j} = 0 \; \forall j \in [n], i \in [2]$.
- We then add one new node s' and one new edge e_{n+1}: For the minimum spanning tree problem, e_{n+1} connects s' to any of the other nodes. For the shortest path problem, the edge e_{n+1} leads from s' to the original start node s.
- We construct cost intervals for the new edge: For some upper bound $B \geq \max_{i=1,2} \sum_{j\in[n]} \hat{c}_{i,j}$ we define $\hat{c}_{(\cdot,n+1)} := (0,0)^T, \delta_{(\cdot,n+1)} := (B,B)^T$.
- We define the new feasible set $\mathcal{X}' := \left\{ \begin{pmatrix} x \\ 1 \end{pmatrix} : x \in \mathcal{X} \right\}$.

Note, that in case of the spanning tree problem, $\mathcal{X}'$ represents the set of all spanning trees in the new graph, since the only edge connecting s' to the old graph is e_{n+1}. In case of the robust shortest path problem, $\mathcal{X}'$ represents the set of all paths from the new node s' to the original destination node t in the new graph, because s' has exactly one outgoing edge e_{n+1}, which ends in the original start node s.
Constructed like this, for every $x \in \mathcal{X}$ the solution $x' := (x,1)^T$ is feasible for I' and for every $x' \in \mathcal{X}'$, the solution $x := (x'_1, ..., x'_n)^T$ is feasible for I. Hence, every feasible solution x for I corresponds to a feasible solution x' for I' and vice versa.
Since for every $x' \in \mathcal{X}'$ we have $x'_{(n+1)} = 1$, its worst case scenario is either

$$\begin{aligned} c^1 &: c^1_{1,(n+1)} = B, c^2_{1,(n+1)} = 0, c^1_{i,j} = \hat{c}_{i,j} \; \forall \; j \neq n+1 \text{ or} \\ c^2 &: c^1_{1,(n+1)} = 0, c_{2,(n+1)} = B, c_{i,j} = \hat{c}_{i,j} \; \forall \; j \neq n+1, \end{aligned}$$

because all other feasible scenarios are equivalent to just considering the nominal edge lengths (since $\Gamma = 1$). The choice of B ensures $z_1(x', c^1) \geqq z_2(x', c^1)$ and $z_2(x', c^2) \geqq z_1(x', c^2)$ for all

$x' \in \mathcal{X}'$. It follows that for every $x' \in \mathcal{X}'$

$$\begin{aligned}\max_{c\in\mathcal{U}} \min_{i=1,2} z_i(x', c) &= \max\left\{\min\{z_1(x', c^1), z_2(x', c^1)\}, \min\{z_1(x', c^2), z_2(x', c^2)\}\right\} \\ &= \max\left\{z_2(x', c^1), z_1(x', c^2)\right\} = \max\left\{\sum_{j\in[n]} \hat{c}_{2,j} x'_j, \sum_{j\in[n]} \hat{c}_{1,j} x'_j\right\} \\ &= \max_{i=1,2} \sum_{j\in[n]} \hat{c}_{i,j} x'_j = \max_{i=1,2} \sum_{j\in[n]} b_{i,j} x'_j.\end{aligned}$$

We conclude that an optimal solution for I' corresponds to an optimal solution for I and vice versa. □

5 Conclusion

In this paper we introduced two methods to find minmax robust efficient solutions based on scalarizations: the min-ordering and the max-ordering method. We have shown that the max-ordering method finds (all) point-based minmax robust weakly efficient solutions. The min-ordering solution finds set-based minmax robust weakly efficient solutions, which cannot necessarily be found with scalarization based methods for multi-objective robust optimization from the literature.

We investigated the resulting scalarized problems (P-min(r, λ)) and (P-max(r, λ)) for multi-objective combinatorial problems with particular uncertainty sets. For interval uncertainty we could show that only one scenario needs to be considered. Then, (P-max(r, λ)) reduces to a single-objective minmax robust problem with discrete uncertainty set, whereas a solution to (P-min(r, λ)) can be found by solving several single-objective deterministic problems with the same feasible set. We further extended the single-objective concept of bounded uncertainty to the multi-objective case. We developed MILP-formulations for both (P-min(r, λ)) and (P-max(r, λ)) with bounded uncertainty and investigated the complexity of the resulting problems.

The first question in mind for further investigations is, how to solve (P-min(r, λ)) and (P-max(r, λ)) in case of multi-objective robust combinatorial problems with other uncertainty sets, e.g., discrete scenarios sets or polyhedral or ellipsoidal uncertainty. Also, the complexity of (P-min(r, λ)) with uncertainty set $\mathcal{U}^c$ remains an open question.

Further research could be done on specialized solution approaches for particular combinatorial problems, for example the shortest path or minimal spanning tree problem. It is also interesting to check if solutions to other robustness concepts, e.g., hull-based minmax robust efficiency [BF17], multi-scenario efficiency [BS16], or lightly robust efficiency [IS16] can be found with the min-ordering or max-ordering method.

A variant of the max-ordering or min-ordering optimization problem is to look for the second/third/... highest or smallest objective instead of the maximum or minimum. Moreover, we have shown that the solutions of (P-min(r, λ)) and (P-max(r, λ)) have quite different properties and characterizations. It would therefore also be of interest to consider a combination of both by choosing any ordered median function as scalarizing function and analyze the resulting problems.

Acknowledgments

Lisa Thom was supported by DFG RTG 1703 “Resource Efficiency in Interorganizational Networks”.

References

[BF17] R. Bokrantz and A. Fredriksson. Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. *European Journal of Operational Research*, 262(2):682–692, 2017.

[BS03] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.

[BS16] M. Botte and A. Schöbel. Dominance for multi-objective robust optimization. Technical Report 2016-8, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2016.

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

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

[BTN98] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[DKW12] E.K. Doolittle, H.L. M. Kerivin, and M. M. Wiecek. A robust multiobjective optimization problem with application to internet routing. Technical Report R2012-11-DKW, Clemson University, 2012.

[Ehr06] M. Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.

[EIS14] M. Ehrgott, J. Ide, and A. Schöbel. Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31, 2014.

[FW14] J. Fliege and R. Werner. Robust multiobjective optimization & applications in portfolio optimization. *European Journal of Operational Research*, 234(2):422–433, 2014.

[HNS13] F. Hassanzadeh, H. Nemati, and M. Sun. Robust optimization for multiobjective programming problems with imprecise information. *Procedia Computer Science*, 17:357–364, 2013.

[Ide14] J. Ide. *Concepts of Robustness for Uncertain Multi-Objective Optimization*. PhD thesis, Universität Göttingen, 2014.

[IS16] J. Ide and A. Schöbel. Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. *OR Spectrum*, 38(1):235–271, 2016.

[KL12] D. Kuroiwa and G. M. Lee. On robust multiobjective optimization. *Vietnam Journal of Mathematics*, 40(2-3):305–317, 2012.

[KRSS16] K. Kuhn, A. Raith, M. Schmidt, and A. Schöbel. Bi-objective robust optimisation. *European Journal of Operational Research*, 252(2):418–431, 2016.

[KY97] P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Boston, 1997.

[RSST17] A. Raith, M. Schmidt, A. Schöbel, and L. Thom. Extensions of labeling algorithms for multi-objective uncertain shortest path problems. Technical Report 2017-2, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Universität Göttingen, 2017.

[RSST18] A. Raith, M. Schmidt, A. Schöbel, and L. Thom. Multi-objective minmax robust combinatorial optimization with cardinality-constrained uncertainty. *European Journal of Operational Research*, 267(2):628–642, 2018.

[Soy73] A. L. Soyster. Technical note—Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

[WD16] M. M. Wiecek and G. M. Dranichak. Robust multiobjective optimization for decision making under uncertainty and conflict. In A. Gupta and A. Capponi, editors, *Optimization Challenges in Complex, Networked and Risky Systems*, INFORMS Tutorials in Operations Research (J.C. Smith, ed.), chapter 4, pages 84–114. INFORMS, 2016.

[WLD+17] L. Wang, Q. Li, R. Ding, M. Sun, and G. Wang. Integrated scheduling of energy supply and demand in microgrids under uncertainty: A robust multi-objective optimization approach. *Energy*, 130:1 – 14, 2017.

Danke

Als erstes möchte ich meiner Doktormutter Anita Schöbel danken: Mit deiner Begeisterung für Mathematik und deiner optimistischen Art hast du mich immer wieder aufs Neue motiviert. Danke für deine Unterstützung und deinen hilfreichen Rat und dafür, dass du trotz deiner vielen Aufgaben immer ansprechbar warst und dir Zeit für meine Anliegen genommen hast!

Bei ihr und meinen beiden anderen Koautorinnen, Andrea Raith und Marie Schmidt, bedanke ich mich außerdem besonders für die gute Zusammenarbeit und die spannenden und ertragreichen Diskussionen. Marie, vielen Dank für die fröhlichen und produktiven Stunden, die wir gemeinsam in Göttingen und Rotterdam verbracht haben, und dafür, dass du dich bereit erklärt hast, meine Arbeit zu begutachten! Andrea, dir möchte ich besonders für die Einladung nach Auckland und unsere Zusammenarbeit dort danken, und für deinen Einsatz für unser Labeling-Paper.

Mein Dank geht auch an meine Zweitbetreuerin Anja Fischer: Danke für deine Hilfsbereitschaft, deine guten Ideen und dein Lachen!

Für die Finanzierung meiner Stelle, meiner Forschungsaufenthalte und Konferenzbesuche bin ich der DFG und dem Graduiertenkolleg 1703 zu Dank verpflichtet. Ich danke auch den Mitgliedern des Kollegs für die Gemeinschaft und den Wissensaustausch, insbesondere der Leiterin Jutta Geldermann, Meike und Kristin für Rat und Tat und ihr Engagement, und Willy für unsere interdisziplinäre Zusammenarbeit.

Ganz besonders möchte ich mich bei meiner Arbeitsgruppe bedanken: für die Gemeinschaft und die angenehme Arbeitsatmosphäre, für viele gemeinsame Mittagessen und selbstgebackene Kuchen, für Beistand in technischen, mathematischen und anderen Fragen des Lebens, für Chorgesang und gemeinsames Forschen. Danke, Alex, Anita, Anja, Corinna, Jörn, Jonas, Julius, Marco Be., Marco Bo., Mirko, Morten, Philine, Robert, Sebastian und Sönke! Ich danke auch Fabian für seine Programmierarbeit und unserem inoffiziellen AG-Mitglied Emelie, deren Lächeln Herzen öffnet.

Ich danke allen Freunden, die mich in meiner Doktorandenzeit unterstützt haben, sei es durch ein offenes Ohr für meine Sorgen oder ihr Ablenkungsvermögen. Besonderer Dank geht an Charlotte und Philine für das Korrekturlesen meiner Arbeit!

Mit ihrer Zuversicht und bedingungslosen Unterstützung hat mir meine Familie in dieser Zeit sehr geholfen. Mama, Papa und Oma, danke, dass ihr für mich da seid!

Hauke, dir gebührt besonderer Dank für deine Geduld, deinen guten Rat und dein offenes Ohr (auch morgens um fünf) und dafür, dass du es immer schaffst, die Welt ein wenig heller aussehen zu lassen. Es tut gut, dich an meiner Seite zu wissen.