European Journal of Operational Research 261 (2017) 421–435

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Continuous Optimization

A biobjective approach to recoverable robustness based on location planning[☆]

Emilio Carrizosa[a], Marc Goerigk[b,*], Anita Schöbel[c]

[a] Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Sevilla, Spain
[b] Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom
[c] Institute for Numerical and Applied Mathematics, Georg-August University, Göttingen, Germany

ARTICLE INFO

Article history:
Received 26 March 2016
Accepted 10 February 2017
Available online 16 February 2017

Keywords:
Robustness & sensitivity analysis
Robust optimization
Location planning
Biobjective optimization

ABSTRACT

Finding robust solutions of an optimization problem is an important issue in practice, and various concepts on how to define the robustness of a solution have been suggested. The idea of recoverable robustness requires that a solution can be recovered to a feasible one as soon as the realized scenario becomes known. The usual approach in the literature is to minimize the objective function value of the recovered solution in the nominal or in the worst case.

As the recovery itself is also costly, there is a trade-off between the recovery costs and the solution value obtained; we study both, the recovery costs and the solution value in the worst case in a biobjective setting.

To this end, we assume that the recovery costs can be described by a metric. We show that in this case the recovery robust problem can be reduced to a location problem.

We show how weakly Pareto efficient solutions to this biobjective problem can be computed by minimizing the recovery costs for a fixed worst-case objective function value and present approaches for the case of linear and quasiconvex problems for finite uncertainty sets. We furthermore derive cases in which the size of the uncertainty set can be reduced without changing the set of Pareto efficient solutions.

1. Introduction

Robust optimization is a popular paradigm to handle optimization problems contaminated with uncertain data, see, e.g., Aissi, Bazgan, and Vanderpooten (2009); Ben-Tal, Ghaoui, and Nemirovski (2009) and references therein. Starting from conservative robustness models requiring that the robust solution is feasible for any of the possible scenarios, new concepts have been developed, see Goerigk and Schöbel (2016) for a recent survey. These concepts allow to relax this conservatism and to control the *price of robustness*, i.e., the loss of objective function value one has to pay in order to obtain a robust solution, see Bertsimas and Sim (2004). In many real-world problems these new robustness concepts have been successfully applied.

Motivated by two-stage stochastic programs, one class of such new models includes the so called *recoverable robustness* introduced in Cicerone, D'Angelo, Stefano, Frigioni, and Navarra (2007); Liebchen, Lübbecke, Möhring, Stiller, and Zaroliagis (2009) and independently also used in Erera, Morales, and Savelsbergh (2009). Recoverable robustness is a two-stage approach that does not require the robust solution to be feasible for all scenarios. Instead, a recoverable-robust solution comes together with a *recovery strategy* which is able to adapt the solution to make it feasible for every scenario. Such a recovery strategy can be obtained by modifying the values of the solution or by allowing another resource or spending additional budget, as soon as it becomes known which scenario occurs. Unfortunately, a recoverable-robust solution can only be determined efficiently for simple problems with special assumptions on the uncertainties and on the recovery algorithms (see Stiller, 2008), and the recoverable-robust counterpart is known to be NP-hard even in simple cases (Cicerone, D'Angelo, Stefano, Frigioni, & Navarra, 2009a).

Our contributions. In this paper we analyze the two main goals in recoverable robustness: Obtaining a good objective function value in the worst case while minimizing the recovery costs. We consider the ε-constrained version as a geometric problem, which allows to interpret robustness as a location planning problem, and derive results on Pareto efficient solutions and how to compute them.

[☆] Partially supported by grants SCHO 1140/3-2 within the DFG programme *Algorithm Engineering*, and grant MTM2012-36163-C06-03.

[*] Corresponding author.
E-mail address: m.goerigk@lancaster.ac.uk (M. Goerigk).

http://dx.doi.org/10.1016/j.ejor.2017.02.014
0377-2217/© 2017 Elsevier B.V. All rights reserved.

Overview. The remainder of the paper is structured as follows.

In the next section we sketch classic and more recent robustness concepts before we introduce the biobjective version of recoverable robustness in Section 3. We then analyze how to solve the scalarization of the recoverable-robust counterpart in Section 4, and consider reduction approaches in Section 5. After discussing numerical experiments in Section 6, we conclude with a summary of results and an outlook to further research in Section 7.

2. Robustness concepts

2.1. Uncertain optimization problems

We consider optimization problems that can be written in the form

$$\text{(P)} \qquad \begin{aligned} &\text{minimize } f(x)\\ &\text{s.t. } F(x) \le 0\\ &x \in \mathcal{X},\end{aligned}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a closed set, $F : \mathbb{R}^n \to \mathbb{R}^m$ describes the m constraints and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized. We assume f and F to be continuous. In practice, the constraints and the objective may both depend on parameters which are in many cases not exactly known. In order to accommodate such uncertainties, the following class of problems is considered instead of (P).

Notation 1. An *uncertain optimization problem* is given as a parameterized family of optimization problems

$$\text{P}(\xi) \qquad \begin{aligned} &\text{minimize } f(x,\xi)\\ &\text{s.t. } F(x,\xi) \le 0\\ &x \in \mathcal{X},\end{aligned}$$

where $F(\cdot,\xi) : \mathbb{R}^n \to \mathbb{R}^m$ and $f(\cdot,\xi) : \mathbb{R}^n \to \mathbb{R}$ are continuous functions for any fixed $\xi \in \mathcal{U}$, $\mathcal{U} \subseteq \mathbb{R}^M$ being the *uncertainty set* which contains all possible scenarios $\xi \in \mathbb{R}^M$ which may occur (see also Ben-Tal et al., 2009).

A scenario $\xi \in \mathcal{U}$ fixes the parameters of f and F. It is often known that all scenarios that may occur lie within a given uncertainty set $\mathcal{U}$, however, it is not known beforehand which of the scenarios $\xi \in \mathcal{U}$ will be realized. We assume that $\mathcal{U}$ is a closed set in $\mathbb{R}^M$ containing at least two elements (otherwise, no uncertainty would affect the problem). Contrary to the setting of stochastic optimization problems, we do not assume a probability distribution over the uncertainty set to be known.

The set $\mathcal{X}$ contains constraints which do not depend on the uncertain parameter ξ. These may be technological or physical constraints on the variables (e.g., some variables represent nonnegative magnitudes, or there are precedence constraints between two events), or may refer to modeling constraints (e.g., some variables are Boolean, and thus they can only take the values 0 and 1).

In short, the uncertain optimization problem corresponding to $\text{P}(\xi)$ is denoted as

$$(\text{P}(\xi), \xi \in \mathcal{U}). \tag{1}$$

We denote

$$\mathcal{F}(\xi) = \{x \in \mathcal{X} : F(x,\xi) \le 0\}$$

as the feasible set of scenario $\xi \in \mathcal{U}$ and

$$f^*(\xi) = \min\{f(x,\xi) : F(x,\xi) \le 0, x \in \mathcal{X}\}$$

as the optimal objective function value for scenario ξ (which might be ∞ if it does not exist). Note that $\mathcal{F}(\xi)$ is closed in $\mathbb{R}^n$, as we assumed $\mathcal{X}$ to be closed, and $F(\cdot,\xi)$ to be continuous. In the following we demonstrate the usage of $\xi \in \mathbb{R}^M$ for the case of linear optimization. In the simplest case, ξ *coincides with the uncertain parameters* of the given optimization problem.

Example 1. Consider a linear program minimize$\{c^t x : Ax \le b, x \in \mathbb{R}^n\}$ with a coefficient matrix $A \in \mathbb{R}^{m,n}$, a right-hand side vector $b \in \mathbb{R}^m$ and a cost vector $c \in \mathbb{R}^n$. If A, b, and c are treated as uncertain parameters, we write

$$\text{P}(A,b,c) \qquad \begin{aligned} &\text{minimize } f(x,(A,b,c)) = c^t x\\ &\text{s.t. } F(x,(A,b,c)) = Ax - b \le 0\\ &x \in \mathcal{X} = \mathbb{R}^n,\end{aligned}$$

i.e., $\xi = (A,b,c) \in \mathbb{R}^M$ with $M = n \cdot m + n + m$

However, in (1) we allow a more general setting, namely that the unknown parameters A, b, c may depend on (other) uncertain parameters $\xi \in \mathbb{R}^M$. For example, there might be $M = 1$ parameter $\xi \in \mathbb{R}$ which determines all values of A, b, c. As an example imagine that the temperature determines the properties of different materials. In such a case we would have

$$f(x,\xi) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}, \text{ and}$$
$$F(x,\xi) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m,$$

where $f(x,\xi) = c(\xi)^t x$ and $F(x,\xi) = A(\xi)x - b(\xi)$.

We now summarize several concepts to handle uncertain optimization problems.

2.2. Strict robustness and less conservative concepts

The first formally introduced robustness concept is called *strict robustness* here. It has been first mentioned by Soyster (1973) and then formalized and analyzed by Ben-Tal, El Ghaoui, and Nemirovski in numerous publications, see Ben-Tal et al. (2009) for an extensive collection of results. A solution $x \in \mathcal{X}$ to the uncertain problem (1) is called *strictly robust* if it is feasible for all scenarios in $\mathcal{U}$, i.e., if $F(x,\xi) \le 0$ for all $\xi \in \mathcal{U}$. The set of strictly robust solutions with respect to the uncertainty set $\mathcal{U}$ is denoted by $\mathbb{SR}(\mathcal{U}) = \bigcap_{\xi\in\mathcal{U}} \mathcal{F}(\xi)$. The *strictly robust counterpart* of (1) is given as

$$\text{RC}(\mathcal{U}) \qquad \begin{aligned} &\text{minimize } \sup_{\xi\in\mathcal{U}} f(x,\xi)\\ &\text{s.t. } x \in \mathbb{SR}(\mathcal{U})\end{aligned}$$

The objective follows the pessimistic view of minimizing the worst case over all scenarios.

Often the set of strictly robust solutions is empty, or all of the strictly robust solutions lead to undesirable solutions (i.e., with considerably worse objective values than a nominal solution would achieve). Recent concepts of robustness hence try to overcome the “over-conservative” nature of the previous approach. In this paper we deal with recoverable robustness which is described in the next section. Other less conservative approaches include the approach of Bertsimas and Sim (2004), reliability (Ben-Tal & Nemirovski, 2000), light robustness (Fischetti & Monaci, 2009; Schöbel, 2014), adjustable robustness (Ben-Tal, Goryashko, Guslitzer, & Nemirovski, 2004) (which will be used in Section 3.3), and comprehensive robustness (Ben-Tal, Boyd, & Nemirovski, 2006). For a more detailed recent overview on different robustness concepts we refer to Goerigk and Schöbel (2016).

3. A biobjective approach to recoverable robustness

Our paper extends the recently published concepts of *recoverable robustness*. As before, we consider a parameterized problem

$$\begin{array}{lll} P(\xi) & \text{minimize } f(x,\xi) \\ & \text{s.t. } F(x,\xi) \le 0 \\ & \quad x \in \mathcal{X} \end{array}$$

The idea of recoverable robustness (see Liebchen et al., 2009) is to allow that a solution can be recovered to a feasible one for every possible scenario. There, a solution $x \in \mathcal{X}$ is called *recoverable-robust* if there is a function $y : \mathcal{U} \to \mathcal{X}$ such that for any possible scenario $\xi \in \mathcal{U}$, the solution $y(\xi) \in \mathcal{F}(\xi)$ is not too different from the original solution x. This includes on the one hand the costs for changing the solution x to the solution $y(\xi)$, and on the other hand the objective function value of $y(\xi)$ compared to the objective function value of x. The solution $y(\xi)$ is called the *recovery solution* for scenario ξ.

Examples include recoverable-robust models for linear programming (Stiller, 2008), shunting (Cicerone et al., 2007), timetabling (Cicerone et al., 2009b), platforming (Caprara, Galli, Stiller, & Toth, 2014), the empty repositioning problem (Erera et al., 2009), railway rolling stock planning (Cacchiani et al., 2012) and the knapsack problem (Büsing, Koster, & Kutschka, 2011). An extensive investigation can be found in Stiller (2008). Note that the model has the drawback that even for simple optimization problems an optimal recoverable-robust solution is usually hard to determine.

3.1. Model formulation

Various goals may be followed when computing a recoverable-robust solution: On the one hand, the new solution should be recoverable to a good solution $y(\xi) \in \mathcal{F}(\xi)$ for every scenario $\xi \in \mathcal{U}$. On the other hand, also the costs of the recovery are important: A new solution has to be implemented, and if x differs too much from $y(\xi)$ this might be too costly. We assume that the recovery costs can be measured by a metric $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. An example for metric recovery costs can be found, e.g., for shunting in Cacchiani et al. (2012); recovery costs defined by norms are also used frequently, e.g., in timetabling (Liebchen et al., 2009), in recoverable-robust linear programming (Stiller, 2008), or in vehicle scheduling problems (Goerigk, Deghdak, & T'Kindt, 2015). Recently, distance functions have also played a role in the development of uncertainty sets for distributional robustness, see, e.g., Ben-Tal, Hertog, De Waegenaere, Melenberg, and Rennen (2013); Bertsimas, Gupta, and Kallus (2013); Carlsson, Behroozi, and Mihic (2015).

In our approach we use both objectives: quality of the recovered solution and recovery costs. The resulting biobjective model for recoverable robustness can be formulated as follows:

$$\begin{array}{ll} \text{(Rec)} & \text{vec} - \text{minimize } (\mathbf{f}(y), \mathbf{r}(x,y)) \\ & = \left(\sup_{\xi \in \mathcal{U}} f(y(\xi),\xi), \sup_{\xi \in \mathcal{U}} d(x, y(\xi)) \right) \\ & \text{s.t. } F(y(\xi),\xi) \le 0 \quad \text{for all } \xi \in \mathcal{U} \\ & x \in \mathcal{X}, y : \mathcal{U} \to \mathcal{X} \end{array}$$

We look for a recoverable robust solution x together with a recovery solution $y(\xi) \in \mathcal{F}(\xi)$ for every scenario $\xi \in \mathcal{U}$. Note that if $\mathcal{U}$ is infinite, (Rec) is not a finite-dimensional problem. In the objective function we consider

- the quality $f(y(\xi), \xi)$ of the recovery solutions, which will finally be implemented, in the worst case, and
- the costs of the recovery $d(x, y(\xi))$, i.e., changing x to $y(\xi)$, again in the worst case.

As usual in multi-criteria optimization we are interested in finding Pareto efficient solutions to this problem. Recall that a solution $(x \in \mathcal{X}, y : \mathcal{U} \to \mathcal{X})$ is weakly Pareto efficient if there does not exist another solution $x' \in \mathcal{X}, y' : \mathcal{U} \to \mathcal{X}$ such that

$$\sup_{\xi \in \mathcal{U}} f(y'(\xi),\xi) < \sup_{\xi \in \mathcal{U}} f(y(\xi),\xi) \text{ and}$$
$$\sup_{\xi \in \mathcal{U}} d(x', y'(\xi)) < \sup_{\xi \in \mathcal{U}} d(x, y(\xi)).$$

If there does not even exist a solution $x' \in X, y' : \mathcal{U} \to \mathcal{X}$ for which one of the two inequalities holds with equality, then (x, y) is called Pareto efficient.

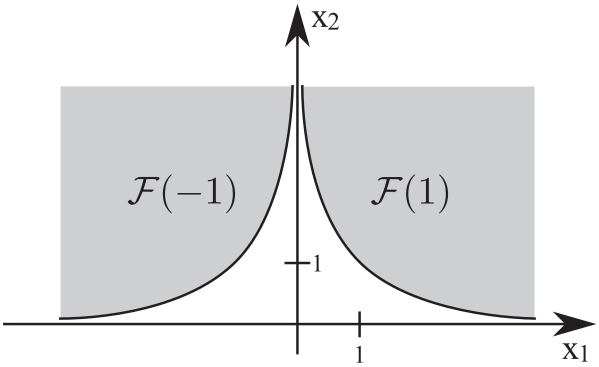


Fig. 1. An instance of (Rec) not having any Pareto efficient solution for an uncertainty set $\mathcal{U}$ with only two scenarios.

Notation 2. We call x *recoverable-robust* for (Rec) if there exists $y : \mathcal{U} \to \mathcal{X}$ such that (x, y) is Pareto efficient for (Rec). $y(\xi)$ is called the recovery solution for scenario ξ.

We are interested in finding recoverable-robust solutions x. Note that (Rec) depends on the uncertainty set $\mathcal{U}$. This dependence is studied in Section 5.

In (Rec), the worst-case objective $\mathbf{f}$ does not depend on x. This is because we assume that x is always modified to the appropriate solution $y(\xi)$ when the scenario is revealed.

We remark, that even if x or y is fixed, the resulting problem (Rec) is still challenging. If x is given, we still have to solve a biobjective problem and choose $y(\xi)$ either with a good objective function value in scenario ξ or with small recovery costs close to x. If y is given, (Rec) reduces to a single-objective problem in which a point is searched which minimizes the maximum distance to all points $y(\xi), \xi \in \mathcal{U}$.

Our first result is negative: Pareto efficient solutions need not exist even for a finite uncertainty set and bounded recovery costs as the following example demonstrates.

Example 2. Consider the uncertain program

$$\begin{array}{lll} P(\xi) & \min f(x,\xi) \\ & \text{s.t. } 1 \le \xi x_1 x_2 \\ & \quad \xi x_1 \ge 0 \\ & \quad x_2 \ge 0, \end{array}$$

where $\mathcal{U} = \{-1, 1\}$ is the uncertainty set and $\mathcal{X} = \mathbb{R}^2$. The feasible sets of scenario $\xi^1 = -1$ and scenario $\xi^2 = 1$ are given by:

$$\mathcal{F}(-1) = \{(x_1,x_2) \in \mathbb{R}^2 : x_1 x_2 \le -1,\ x_1 \le 0,\ x_2 \ge 0\},$$
$$\mathcal{F}(1) = \{(x_1,x_2) \in \mathbb{R}^2 : x_1 x_2 \ge 1,\ x_1, x_2 \ge 0\}.$$

Both feasible sets are depicted in Fig. 1.

For the objective function $f(x,\xi) = |x_1|$ and the recovery distance $d(x,y) = \|y - x\|_1$ the problem does not have any Pareto efficient solution.

This can be seen as follows: Let $x = (x_1, x_2)$, $y^1 = (y_1^1, y_2^1)$ and $y^{-1} = (y_1^{-1}, y_2^{-1})$ be an arbitrary feasible solution to (Rec) where y^1 and y^{-1} are the two recovery solutions for the scenarios $\xi = 1$

and $\xi = -1$. *Let* $a := \max\{x_2, y^1_2, y^{-1}_2\} + 1$. *Then the solution* $\bar{x} := (0, a), \bar{y}^1 := (\frac{1}{a}, a)$ *and* $\bar{y}^{-1} := (-\frac{1}{a}, a)$ *is strictly better in both objective functions. To this end we use that* $y^1_1 \geq \frac{1}{y^1_2} > \frac{1}{a}$ *and* $y^{-1}_1 \leq -\frac{1}{y^{-1}_2} < -\frac{1}{a}$ *and compute*

$$\begin{aligned}
\mathbf{f}(y^1, y^{-1}) &= \max\{|y^1_1|, |y^{-1}_1|\} > \max\left\{\left|\tfrac{1}{a}\right|, \left|-\tfrac{1}{a}\right|\right\} = \mathbf{f}(\bar{y}^1, \bar{y}^{-1}),\\
\mathbf{r}(x, y^1, y^{-1}) &= \max\{\|x - y^1\|_1, \|x - y^{-1}\|_1\}\\
&\geq \max\{|x_1 - y^1_1|, |x_1 - y^{-1}_1|\}\\
&\geq \frac{|y^1_1 - y^{-1}_1|}{2} > \frac{1}{a} = \max\{\|\bar{x} - \bar{y}^1\|_1, \|\bar{x} - \bar{y}^{-1}\|_1\}\\
&= \mathbf{r}(x, \bar{y}^1, \bar{y}^{-1}).
\end{aligned}$$

It is known that all weakly Pareto efficient solutions are optimal solutions of one of the two ε-constraint scalarizations which are given by bounding one of the objective functions while minimizing the other one.

The first scalarization bounds the recovery costs and minimizes the objective function value in the first place, i.e.,

$$\begin{aligned}
(\mathrm{Rec}^{\mathrm{class}}(\delta)) \qquad & \text{minimize } \sup\nolimits_{\xi \in \mathcal{U}} f(y(\xi), \xi)\\
\text{s.t. } & d(x, y(\xi)) \leq \delta \quad \text{for all } \xi \in \mathcal{U}\\
& F(y(\xi), \xi) \leq 0 \quad \text{for all } \xi \in \mathcal{U}\\
& x \in \mathcal{X}, y : \mathcal{U} \to \mathcal{X}.
\end{aligned}$$

This problem has been introduced as *recoverable robustness* (see Liebchen et al., 2009) and solved in several special cases, e.g., in Büsing et al. (2011); Goerigk et al. (2015); Kasperski and Zielinski (2015). It is hence denoted as the *classic* scalarization approach.

In this paper we look at the other scalarization in which we minimize the recovery costs while requiring a minimal quality of the recovery solutions:

$$\begin{aligned}
(\mathrm{Rec}(\varepsilon)) \qquad & \text{minimize } \sup\nolimits_{\xi \in \mathcal{U}} d(x, y(\xi))\\
\text{s.t. } & f(y(\xi), \xi) \leq \varepsilon \quad \text{for all } \xi \in \mathcal{U} \qquad (2)
\end{aligned}$$

$$\begin{aligned}
& F(y(\xi), \xi) \leq 0 \quad \text{for all } \xi \in \mathcal{U} \qquad (3)\\
& x \in \mathcal{X}, y : \mathcal{U} \to \mathcal{X}
\end{aligned}$$

Note that Constraints (2) and (3) of this second scalarization do *not* depend on *x*. To determine feasibility of (Rec(ε)), we hence check if for every $\xi \in \mathcal{U}$ there exists $y(\xi)$ such that

$$f(y(\xi), \xi) \leq \varepsilon \text{ and } F(y(\xi), \xi) \leq 0,$$

i.e., if the sets

$$\begin{aligned}
\mathcal{G}_\varepsilon(\xi) &:= \{y \in \mathcal{X} : F(y, \xi) \leq 0 \text{ and } f(y, \xi) \leq \varepsilon\}\\
&= \mathcal{F}(\xi) \cap \{y \in \mathcal{X} : f(y, \xi) \leq \varepsilon\}
\end{aligned}$$

are not empty for all $\xi \in \mathcal{U}$. For a fixed scenario ξ the set $\mathcal{G}_\varepsilon(\xi)$ contains all feasible recovery solutions *y* which have a sufficient quality, i.e., it is the intersection of the feasible set with the level set (for level ε) of the problem $P(\xi)$. To extend a given *x* to a feasible solution, we choose some $y(\xi) \in \mathcal{G}_\varepsilon(\xi)$ which is closest to *x* w.r.t the metric *d*. This is possible since $\mathcal{G}_\varepsilon(\xi)$ is closed: we define

$$d(x, \mathcal{G}_\varepsilon(\xi)) = \min_{y \in \mathcal{G}_\varepsilon(\xi)} d(x, y),$$

where the minimum exists whenever $\mathcal{G}_\varepsilon(\xi) \neq \emptyset$.

With $d(x, \emptyset) := \infty$, we can now rewrite (Rec(ε)) to an equivalent problem in the (finitely many) *x*-variables only:

$$\begin{aligned}
(\mathrm{Rec}'(\varepsilon)) \qquad & \text{minimize } \sup\nolimits_{\xi \in \mathcal{U}} d(x, \mathcal{G}_\varepsilon(\xi)) \qquad (4)\\
& \text{s.t. } x \in \mathcal{X},
\end{aligned}$$

i.e., *x* is an optimal solution to (Rec'(ε)) if and only if (*x*, *y*) with $y(\xi) \in \operatorname{argmin}_{\mathcal{G}_\varepsilon(\xi)} d(x, y)$ is optimal for (Rec(ε)).

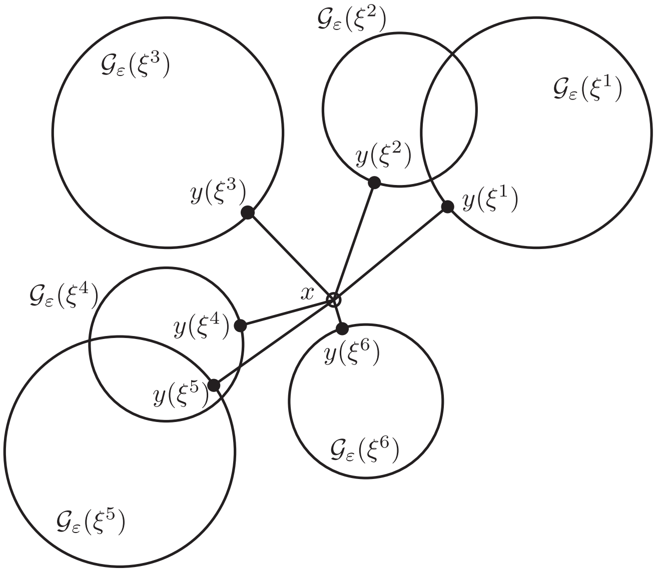

$\mathcal{G}_\varepsilon(\xi^2)$
$\mathcal{G}_\varepsilon(\xi^3)$
$\mathcal{G}_\varepsilon(\xi^1)$
$y(\xi^2)$
$y(\xi^3)$
$y(\xi^1)$
$\mathcal{G}_\varepsilon(\xi^4)$
x
$y(\xi^4)$
$y(\xi^6)$
$y(\xi^5)$
$\mathcal{G}_\varepsilon(\xi^6)$
$\mathcal{G}_\varepsilon(\xi^5)$

Fig. 2. The sets $\mathcal{G}_\varepsilon(\xi), \xi \in \mathcal{U}$ together with an optimal solution *x* and recovery solutions $y(\xi), \xi \in \mathcal{U}$. *x* minimizes the maximum distance to the sets $\mathcal{G}_\varepsilon(\xi)$.

3.2. Location-based interpretation of (Rec(ε))

In a classic location problem (known as the *Weber problem* or as the *Fermat-Torricelli problem*, see e.g., Drezner, Klamroth, Schöbel, & Wesolowsky (2002)) we have given a set of points, called *existing facilities*, and we look for a new point minimizing a measure of distance to these given points. If the distance to the farthest point is considered as the objective function, the problem is called *center location problem*. We have already seen that for given $y : \mathcal{U} \to \mathcal{X}$, our biobjective problem (Rec) reduces to the problem of finding a location *x* which minimizes the maximum distance to the set $\{y(\xi) : \xi \in \mathcal{U}\}$, i.e., a classic center location problem.

We now show that also the ε-constrained version (Rec(ε)) of recoverable robustness

$$\min_{x \in \mathcal{X}} \max_{\xi \in \mathcal{U}} d(x, \mathcal{G}_\varepsilon(\xi))$$

can be interpreted as the following location problem: The existing facilities are not points but the sets $\mathcal{G}_\varepsilon(\xi), \xi \in \mathcal{U}$. (Rec($\varepsilon$)) looks for a new location in the metric space $\mathcal{X}$, namely a point $x \in \mathcal{X}$ which minimizes the maximum distance to the given sets. For a finite uncertainty set $\mathcal{U}$, such location problems have been studied in Brimberg and Wesolowsky (2000,2002a) for the center objective function and in Brimberg and Wesolowsky (2002b); Nickel, Puerto, and Rodriguez-Chia (2003) for median or ordered median objective functions. We adapt the notation of location theory and call such a point (which then is an optimal solution to (Rec(ε)) a *center* with respect to $\{\mathcal{G}_\varepsilon(\xi) : \xi \in \mathcal{U}\}$ and the distance function *d*. In our further analysis we consider (Rec(ε)) from a location's point of view. To this end, let us denote the objective function of (Rec(ε)) by

$$r_\varepsilon(x, \mathcal{U}) = \sup_{\xi \in \mathcal{U}} d(x, \mathcal{G}_\varepsilon(\xi))$$

and let us call $r_\varepsilon(x, \mathcal{U})$ the (recovery) radius of *x* with respect to ε and $\mathcal{U}$. Let $r^*_\varepsilon(\mathcal{U})$ denote the best possible recovery radius over $\mathcal{X}$ (if it exists). For a center location x^* we then have $r_\varepsilon(x^*, \mathcal{U}) = r^*_\varepsilon(\mathcal{U})$. Fig. 2 illustrates this point of view: For six scenarios, we depicted the sets $\mathcal{G}_\varepsilon(\xi_i)$, $i = 1, \ldots, 6$. We look for a point *x* which minimizes the maximum distance to these sets. If *x* has been fixed, the recovery solutions $y^i := y(\xi^i), i = 1, \ldots, 6$ can be chosen as projections from *x* on the sets $\mathcal{G}_\varepsilon(\xi^i)$.

For specific shapes of the sets $\mathcal{G}_\varepsilon(\xi)$, algorithms of location theory may be used to find a solution *x* to (*Rec*(ε)) efficiently. For example, if the sets $\mathcal{G}_\varepsilon(\xi)$ are balls of equal radius, it is known that the point x minimizing the maximum distance to the sets $\mathcal{G}_\varepsilon(\xi)$ is the same as the point x minimizing the maximum distance to the center points of these sets. The latter is a classic point location problem which can be solved efficiently (Megiddo, 1984), hence in such a case an optimal solution to (Rec(ε)) and therefore a weakly Pareto efficient solution for (Rec) can be computed efficiently.

Closely related are quadratic programming problems of the type

$$P(\xi) \quad \min\left\{\sum_{i=1}^{l}\sum_{j=1}^{n}(x_j - \xi_{ij})^2, x \in \mathbb{R}^n\right\}$$

with uncertain parameters $\xi = (\xi_{ij})_{i=1,\dots,l,j=1,\dots,n} \in \mathbb{R}^{l,n}$, where the level sets of the optimization problem P(ξ^k) and hence the sets $\mathcal{G}_\varepsilon(\xi^k)$ for a specific scenario ξ^k are balls around the centroid $\bar{\xi}^k := \frac{1}{l}(\sum_{i=1}^{l}\xi_{i1}^k, \dots, \sum_{i=1}^{l}\xi_{in}^k) \in \mathbb{R}^n$ of the row vectors of ξ^k.

3.3. Relation of the biobjective model to other robustness concepts

We first point out the relation between (Rec) and the concept of strict robustness of Ben-Tal et al. (2009). To this end recall from Section 2.2 that $\mathcal{SR}(\mathcal{U}) = \{x \in \mathcal{X} : F(x, \xi) \leq 0 \text{ for all } \xi \in \mathcal{U}\}$ is the set of strictly robust solutions and RC($\mathcal{U}$) is the strictly robust counterpart of $(P(\xi), \xi \in \mathcal{U})$.

Lemma 1. *Let an uncertain problem* (P(ξ), $\xi \in \mathcal{U}$) *be given. Then we have:*

1. *If $\bar{x}$ is an optimal solution to RC($\mathcal{U}$) then $(\bar{x}, \bar{y})$ with $\bar{y}(\xi) := \bar{x}$* for all *$\xi \in \mathcal{U}$ is a lexicographically minimal solution to (Rec) w.r.t (**r**(x, y), **f**(y)).*
2. *Let $(\bar{x}, \bar{y})$ be a lexicographically minimal solution to (Rec) w.r.t (**r**(x, y), **f**(y)). Then $\mathcal{SR}(\mathcal{U}) \neq \emptyset$ if and only if $\mathbf{r}(\bar{x}, \bar{y}) = 0$ and in this case $(\bar{x}, \bar{y})$ is optimal to RC($\mathcal{U}$).*

Proof.

1. Let $\bar{x}$ be an optimal solution to RC($\mathcal{U}$). Define $\bar{y}(\xi) := \bar{x}$ for all $\xi \in \mathcal{U}$. Then $\mathbf{r}(\bar{x}, \bar{y}) = 0$. Now assume $(\bar{x}, \bar{y})$ is not lexicographically minimal. Then there exists (x', y') with $\mathbf{r}(x', y') = 0$ and $\mathbf{f}(y') < \mathbf{f}(\bar{y})$. The first condition yields that $d(x', y'(\xi)) = 0$ for all $\xi \in \mathcal{U}$, hence $x' = y'(\xi)$ for all $\xi \in \mathcal{U}$, and $x' \in \mathcal{SR}(\mathcal{U})$. Using $x' = y'(\xi)$, the second condition implies $\sup_{\xi\in\mathcal{U}} f(x', \xi) < \sup_{\xi\in\mathcal{U}} f(\bar{x}, \xi)$, a contradiction to the optimality of $\bar{x}$ for RC($\mathcal{U}$).
2. Now let $(\bar{x}, \bar{y})$ be lexicographically minimal to (Rec).
   - Let $\mathbf{r}(\bar{x}, \bar{y}) = 0$. Then $0 = \mathbf{r}(\bar{x}, \bar{y}) = \sup_{\xi\in\mathcal{U}} d(\bar{x}, \bar{y}(\xi))$, i.e., $\bar{x} = \bar{y}(\xi)$ for all $\xi \in \mathcal{U}$. Hence $\bar{x} \in \mathcal{F}(\xi)$ for all $\xi \in \mathcal{U}$, i.e., $\bar{x} \in \mathcal{SR}(\mathcal{U})$.
   - On the other hand, if $\mathcal{SR}(\mathcal{U}) \neq \emptyset$ there exists $x \in \mathcal{F}(\xi)$ for all $\xi \in \mathcal{U}$. We define $y(\xi) := x$ for all $\xi \in \mathcal{U}$ and obtain $\mathbf{r}(x, y) = 0$. Since $(\bar{x}, \bar{y})$ is lexicographically minimal this implies $\mathbf{r}(\bar{x}, \bar{y}) = 0$.

   Finally, if $\mathbf{r}(\bar{x}, \bar{y}) = 0$ we already know that $\bar{x} = \bar{y}(\xi)$ for all $\xi \in \mathcal{U}$ and $\bar{x} \in \mathcal{SR}(\mathcal{U})$, i.e., feasible for RC($\mathcal{U}$). The lexicographic optimality then guarantees that $\bar{x}$ is an optimal solution to RC($\mathcal{U}$). □

Sorting the criteria in the objective function in the other order, i.e., minimizing first **f**(y) and then **r**(x, y) is not directly related to any known robustness concept. This lexicographically minimal solution (x, y) realizes an optimal solution $y(\xi)$ in every scenario, and among these optimal solutions minimizes the recovery costs.

Lemma 2. *Let (x, y) be a solution to (Rec) which is lexicographically minimal w.r.t (**f**(y), **r**(x, y)). Then $f(y(\xi), \xi) = f^*(\xi)$ for all $\xi \in \mathcal{U}$.*

We now turn our attention to (Rec(ε)) and show that this scalarization can be interpreted as adjustable robustness as in Ben-Tal et al. (2004). Motivated by stochastic programming, the variables in this concept are decomposed into two sets: The values for the *here-and-now variables* have to be found in the robust optimization algorithm while the decision about the *wait-and-see variables* can wait until the actual scenario $\xi \in \mathcal{U}$ becomes known. For an uncertain problem (P(ξ), $\xi \in \mathcal{U}$), recall that (Rec(ε)) is given as

$$\min_{x\in\mathcal{X}} \sup_{\xi\in\mathcal{U}} d(x, \mathcal{G}_\varepsilon(\xi)).$$

We can rewrite this problem in the following way:

$$\min_{z,x} \{z : \forall\xi \in \mathcal{U}\ \exists y \in \mathcal{G}_\varepsilon(\xi) : d(x, y) \leq z\}$$

which has the same structure as an adjustable robust problem. As an example, for a problem with linear objective function $f(x, \xi) = c(\xi)^t x$, linear constraints $F(x, \xi) = A(\xi)x - b(\xi) \leq 0$, and $\|\cdot\|_1$ as recovery norm, we may write

$$\min_{z',x}\left\{\sum_{i=1}^{n} z'_i : \forall\xi \in \mathcal{U}\ \exists y : c(\xi)^t y \leq \varepsilon,\ A(\xi)y \leq b(\xi), \right.$$
$$\left. -z' \leq x - y \leq z'\right\}. \tag{5}$$

Note that this is a problem *without* fixed recourse (i.e., coefficients of second-stage variables are affected by uncertainty), such that most of the results in Ben-Tal et al. (2004) are *not* applicable. However, we are still able to apply their results on using heuristic, affinely adjustable counterparts, and Theorem 2.1 from Ben-Tal et al. (2004):

Theorem 1. *Let* (P(ξ), $\xi \in \mathcal{U}$) *be an uncertain linear optimization problem, and let the uncertainty be constraint-wise. Furthermore, let there be a compact set C such that $\mathcal{F}(\xi) \subseteq C$ for all $\xi \in \mathcal{U}$. Then, (Rec(ε)) is equivalent to the following problem*

$$\min_{z',x}\left\{\sum_{i=1}^{n} z'_i : \exists y\ \forall\xi \in \mathcal{U} : c(\xi)^t y \leq \varepsilon,\ A(\xi)y \leq b(\xi), \right.$$
$$\left. -z' \leq x - y \leq z'\right\}. \tag{6}$$

Note that problem (6) is a strictly robust problem, which is considerably easier to solve than problem (5). Furthermore, Ben-Tal et al. (2004) show that there is a semidefinite program for ellipsoidal uncertainty sets which is equivalent to problem (5).

Problem (Rec(ε)) can also be interpreted as a strictly robust problem in x (see (4)). However, the function $\xi \mapsto d(x, \mathcal{G}_\varepsilon)$ has in general not much properties such that most of the known results cannot be directly applied. Nevertheless, our geometric interpretation gives rise to the results of the next section, in particular within the biobjective setting.

4. Solving (Rec(ε))

In this section we investigate the new scalarization (Rec(ε)). After a more general analysis of this optimization problem in Section 4.1, we turn our attention to the case of a finite uncertainty set in Section 4.2 where we consider problems with convex and with linear constraints.

4.1. Analysis of (Rec(ε))

Let us now describe some general properties of problem (Rec(ε)). Since d is a metric we know that

$$0 \leq r_\varepsilon(x, \mathcal{U}) \leq +\infty \quad \text{for all } x \in \mathbb{R}^n, \tag{7}$$

hence the optimal value of (Rec(ε)) is bounded by zero from below, although it is $+\infty$ if all points x have infinite radius $r_\varepsilon(x, \mathcal{U})$. This event may happen even when all sets $\mathcal{G}_\varepsilon(\xi)$ are non-empty. Indeed, consider, for instance, $\mathcal{X} = \mathbb{R}$, $\mathcal{G}_\varepsilon(\xi) = \{\xi\}$ for all $\xi \in \mathcal{U} = \mathbb{R}$. One has, however, that finiteness of $r_\varepsilon(x, \mathcal{U})$ at one point x_0 and one ε implies finiteness of $r_{\varepsilon'}(x, \mathcal{U})$ for all $x \in \mathcal{X}$ and for all $\varepsilon' \geq \varepsilon$. In that case we obtain Lipschitz-continuity of the radius, as shown in the following result.

Lemma 3. *Let an uncertain optimization problem* $(\mathrm{P}(\xi), \xi \in \mathcal{U})$ *be given. Suppose there exists* $x_0 \in \mathbb{R}^n$ *such that* $r_\varepsilon(x_0, \mathcal{U}) < +\infty$. *Then,* $r_\varepsilon(x, \mathcal{U}) < +\infty$ *for all* $x \in \mathbb{R}^n$ *and for all* $\varepsilon' \geq \varepsilon$. *In such a case, the function* $\mathbb{R}^n \ni x \longmapsto r_{\varepsilon'}(x, \mathcal{U})$ *is Lipschitz-continuous with Lipschitz constant* $L = 1$ *for every* $\varepsilon' \geq \varepsilon$.

Proof. Take $x \in \mathbb{R}^n$ and $\xi \in \mathcal{U}$. Let $y \in \mathcal{G}_\varepsilon(\xi)$ such that $d(x_0, \mathcal{G}_\varepsilon(\xi)) = d(x_0, y)$. We have that

$$d(x, \mathcal{G}_\varepsilon(\xi)) \leq d(x, y) \leq d(x, x_0) + d(x_0, y) = d(x, x_0) + d(x_0, \mathcal{G}_\varepsilon(\xi))$$

Hence,

$$\max_{\xi \in \mathcal{U}} d(x, \mathcal{G}_\varepsilon(\xi)) \leq d(x, x_0) + \max_{\xi \in \mathcal{U}} d(x_0, \mathcal{G}_\varepsilon(\xi)) < +\infty,$$

and therefore, $r_\varepsilon(x, \mathcal{U})$ is finite everywhere. Since $r_{\varepsilon'}(x, \mathcal{U}) \leq r_\varepsilon(c, \mathcal{U})$ for all $\varepsilon' \geq \varepsilon$ we also have finiteness if we increase ε.

We now show that $r_\varepsilon(\cdot, \mathcal{U})$ is also Lipschitz-continuous. Let $\delta > 0$, and let $x, x' \in \mathbb{R}^n$. Take ξ^* such that

$$\delta + d(x, \mathcal{G}_\varepsilon(\xi^*)) \geq r_\varepsilon(x, \mathcal{U}).$$

Since $\mathcal{G}_\varepsilon(\xi^*)$ is closed, take also $y' \in \mathcal{G}_\varepsilon(\xi^*)$ such that $d(x', \mathcal{G}_\varepsilon(\xi^*)) = d(x', y')$.

Then,

$$\begin{aligned} r_\varepsilon(x, \mathcal{U}) - r_\varepsilon(x', \mathcal{U}) &\leq \delta + d(x, \mathcal{G}_\varepsilon(\xi^*)) - d(x', \mathcal{G}_\varepsilon(\xi^*)) \\ &\leq \delta + d(x, y') - d(x', y') \\ &\leq \delta + d(x, x'). \end{aligned}$$

Since this inequality holds for any $\delta > 0$, we obtain $r_\varepsilon(x, \mathcal{U}) - r_\varepsilon(x', \mathcal{U}) \leq d(x, x')$, hence the function $r(\cdot, \mathcal{U})$ is Lipschitz-continuous with Lipschitz constant 1. □

In what follows we assume finiteness of the optimal value of (Rec(ε)), and thus Lipschitz-continuity of $r_\varepsilon(\cdot, \mathcal{U})$. Hence, (Rec($\varepsilon$)) may be solved by using standard Lipschitz optimization methods (Sergeyev & Kvasov, 2010).

For a given $x \in \mathbb{R}^n$ let us call $\xi \in \mathcal{U}$ a *worst-case scenario* with respect to x (and $\mathcal{U}$) if

$$d(x, \mathcal{G}_\varepsilon(\xi)) = r_\varepsilon(x, \mathcal{U})$$

and let $WC_\varepsilon(x, \mathcal{U})$ be the set of all worst-case scenarios, i.e., scenarios $\xi \in \mathcal{U}$ yielding the maximal recovery distance for the solution x. Under certain assumptions, any optimal solution x^* to (Rec(ε)) has a set $WC_\varepsilon(x^*, \mathcal{U})$ with at least two elements, as shown in the following result.

Lemma 4. *Let an uncertain optimization problem* $(\mathrm{P}(\xi), \xi \in \mathcal{U})$ *be given. Suppose that* $\mathcal{U}$ *is finite (with at least two elements) and* $\mathcal{X} = \mathbb{R}^n$. *Fix some* ε *and assume that (Rec(ε)) attains its optimum at some* $x^* \in \mathbb{R}^n$. *Then,* $|WC_\varepsilon(x^*, \mathcal{U})| \geq 2$.

Proof. Finiteness of $\mathcal{U}$ implies that the maximum of $d(x^*, \mathcal{G}_\varepsilon(\xi))$ must be attained at some ξ. Hence, $|WC_\varepsilon(x^*, \mathcal{U})| \geq 1$.

In the case that $r_\varepsilon(x^*, \mathcal{U}) = 0$, we have that $WC_\varepsilon(x^*, \mathcal{U}) = \mathcal{U}$. Thus, let $r_\varepsilon(x^*, \mathcal{U}) > 0$.

In the case that $WC_\varepsilon(x^*, \mathcal{U}) = \{\xi^*\}$ for only one scenario $\xi^* \in \mathcal{U}$, we can construct a contradiction by finding a different x with a better radius: Take $y^* \in \mathcal{G}_\varepsilon(\xi^*)$ such that $d(x^*, y^*) = d(x^*, \mathcal{G}_\varepsilon(\xi^*))$, and, for $\lambda \in [0, 1]$, define x_λ as

$$x_\lambda = (1 - \lambda)x^* + \lambda y^*.$$

Since, by assumption, $WC_\varepsilon(x^*, \mathcal{U}) = \{\xi^*\}$ and $\mathcal{U}$ is finite, there exists $\delta > 0$ such that

$$d(x^*, \mathcal{G}_\varepsilon(\xi)) < d(x^*, \mathcal{G}_\varepsilon(\xi^*)) - \delta \qquad \forall \xi \in \mathcal{U},\ \xi \neq \xi^*.$$

Let us show that, for λ close to zero, x_λ has a strictly better objective value than x^*, which would be a contradiction. First we have

$$\begin{aligned} d(x_\lambda, \mathcal{G}_\varepsilon(\xi^*)) &\leq d(x_\lambda, y^*) \\ &= (1 - \lambda)\|x^* - y^*\| = (1 - \lambda)d(x^*, \mathcal{G}_\varepsilon(\xi^*)) \\ &< d(x^*, \mathcal{G}_\varepsilon(\xi^*)) \text{ for } \lambda > 0. \end{aligned}$$

For the remaining scenarios $\xi \neq \xi^*$,

$$\begin{aligned} d(x_\lambda, \mathcal{G}_\varepsilon(\xi)) &\leq \inf_{y \in \mathcal{G}_\varepsilon(\xi)} \left\{ \|x_\lambda - x^*\| + \|x^* - y\| \right\} \\ &= \inf_{y \in \mathcal{G}_\varepsilon(\xi)} \left\{ \lambda\|x^* - y^*\| + \|x^* - y\| \right\} \\ &= \lambda\|x^* - y^*\| + d(x^*, \mathcal{G}_\varepsilon(\xi)) \\ &< \lambda\|x^* - y^*\| + d(x^*, \mathcal{G}_\varepsilon(\xi^*)) - \delta \\ &< d(x^*, \mathcal{G}_\varepsilon(\xi^*)) \text{ for } \lambda < \frac{\delta}{\|x^* - y^*\|}. \end{aligned}$$

Hence, for $0 < \lambda < \frac{\delta}{\|x^* - y^*\|}$, we would have that

$$\max_{\xi \in \mathcal{U}} d(x_\lambda, \mathcal{G}_\varepsilon(\xi)) < d(x^*, \mathcal{G}_\varepsilon(\xi^*)) = \max_{\xi \in \mathcal{U}} d(x^*, \mathcal{G}_\varepsilon(\xi)),$$

contradicting the optimality of x^*. □

If the finiteness assumption of Lemma 4 is dropped, not much can be said about the cardinality of $WC_\varepsilon(x, \mathcal{U})$, since this set can be empty or a singleton:

Example 3. Let $\mathcal{U} = \{-1, 1\} \times [1, \infty)$, and let $F(x, (\xi_1, \xi_2)) = (x - \xi_1)(\xi_2 x - \xi_1\xi_2 + \xi_1)$. Let $f(x) = const$ and choose $\varepsilon > const$. It is easily seen that

$$\begin{aligned} \mathcal{G}_\varepsilon(-1, \xi_2) &= \mathcal{F}(-1, \xi_2) = \left[-1, -1 + \frac{1}{\xi_2}\right] \\ \mathcal{G}_\varepsilon(1, \xi_2) &= \mathcal{F}(1, \xi_2) = \left[1 - \frac{1}{\xi_2}, 1\right] \end{aligned} \tag{8}$$

For $x = 0$, $r_\varepsilon(x, \mathcal{U}) = 1$, but there is no $\xi \in \mathcal{U}$ with $d(x, \mathcal{G}_\varepsilon(\xi)) = 1$. In other words, $WC_\varepsilon(0, \mathcal{U}) = \emptyset$.

4.2. Solving (Rec(ε)) for a finite uncertainty set $\mathcal{U}$

In this section we assume that $\mathcal{U}$ is finite, $\mathcal{U} = \{\xi^1, \ldots, \xi^N\}$. This simplifies the analysis, since we can explicitly search for a solution $y^k = y(\xi^k)$ for every scenario $\xi^k \in \mathcal{U}$. Using the y^k as variables we may formulate (Rec(ε)) as

$$\begin{aligned} \min\ & r \\ \text{s.t.}\ & F(y^k, \xi^k) \leq 0 \text{ for all } k = 1, \ldots, N \\ & f(y^k, \xi^k) \leq \varepsilon \text{ for all } k = 1, \ldots, N \\ & d(x, y^k) \leq r \text{ for all } k = 1, \ldots, N \\ & x \in \mathcal{X}, r \in \mathbb{R} \\ & y^k \in \mathcal{X} \qquad \text{for all } k = 1, \ldots, N. \end{aligned} \tag{9}$$

We can write (Rec(ε)) equivalently as

$$\min_{x \in \mathcal{X}} \max_{1 \leq k \leq N} d(x, \mathcal{G}_\varepsilon(\xi^k)).$$

Assuming that the distance used is the Euclidean $d_2(\cdot, \cdot)$, the function $x \longmapsto \max_k d_2(x, \mathcal{G}_\varepsilon(\xi^k))$ is known to be *d.c.* for closed sets $\mathcal{G}_\varepsilon$ (Horst & Thoai, 1999), i.e., it can be written as a difference of two convex functions, and then the powerful tools of d.c. programming may be used to find a globally optimal solution if (Rec(ε)) is low-dimensional (Blanquero, Carrizosa, & Hansen, 2009), or to design heuristics for more general cases (An & Tao, 2005).

4.2.1. Convex programming problems

We start with optimization problems P(ξ) that have convex sets $\mathcal{G}_\varepsilon(\xi)$ for all $\xi \in \mathcal{U}$. This is the case if the functions F and f of P(ξ) are quasiconvex for all fixed scenarios ξ, and $\mathcal{X}$ is convex. We furthermore assume that d is convex, which is the case, e.g., when d has been derived from a norm, i.e. $d(x, y) = \|y - x\|$ for some norm $\|\cdot\|$.

Let us fix ξ. Then the function $\mathbb{R}^n \ni x \longmapsto d(x, \mathcal{F}(\xi))$ describes the distance between a point and a convex set and is hence convex. We conclude that also $r_\varepsilon(x, \mathcal{U})$ is convex, being the maximum of a finite set of convex functions.

Lemma 5. *Consider an uncertain optimization problem* (P(ξ), $\xi \in \mathcal{U}$) *with quasiconvex objective function $f(\cdot, \xi)$ and quasiconvex constraints $F(\cdot, \xi)$ for any fixed ξ. Let $\mathcal{X} \subseteq \mathbb{R}^n$ be convex, $\mathcal{U}$ be a finite set and d be convex. Then problem (Rec(ε)) is a convex optimization problem.*

In order to solve (Rec(ε)) one can hence apply algorithms suitable for convex programming, e.g., subgradient or bundle methods (Hiriart-Urruty & Lemaréchal, 1993; Sun & Yuan, 2006). In particular, if (Rec(ε)) is unconstrained in x, a necessary and sufficient condition for a point x^* to be an optimal solution is

$$0 \in \partial(r_\varepsilon(x^*, \mathcal{U})),$$

i.e., if 0 is contained in the subdifferential of $r_\varepsilon(\cdot, \mathcal{U})$ at the point x^*. By construction of $r_\varepsilon(\cdot, \mathcal{U})$, we obtain

$$0 \in \operatorname{conv}\{\partial d(x^*, \mathcal{G}_\varepsilon(\xi)) : \xi \in WC_\varepsilon(x^*, \mathcal{U})\}$$

where $WC_\varepsilon(x^*, \mathcal{U})$ is the set of worst-case scenarios (see Hiriart-Urruty & Lemaréchal, 1993), and $\partial d(x^*, \mathcal{G}_\varepsilon(\xi))$ is the subdifferential of $d(\cdot, \mathcal{G}_\varepsilon(\xi))$ at x^*.

Now, $\partial d(x^*, \mathcal{G}_\varepsilon(\xi))$ can be written in terms of the subdifferential of the distance used, see Carrizosa and Fliege (2002), where also easy representations for polyhedral norms or the Euclidean norm are presented. Although we do not know much a priori about the number of worst-case scenarios, we do not need to investigate all possible subsets but may restrict our search to sets which do not have more than $n + 1$ elements as is shown in our next result. This may be helpful in problems with a large number of scenarios but low dimension n for the decisions.

Theorem 2. *Let $\mathcal{U}$ be finite with cardinality of at least $n + 1$. Let $\mathcal{X} = \mathbb{R}^n$. Suppose (Rec($\varepsilon$)) attains its optimum at some x^*, and that for each ξ the functions $F(\cdot, \xi)$ and $f(\cdot, \xi)$ are quasiconvex. Let d be convex. Then there exists a subset $\overline{\mathcal{U}} \subseteq \mathcal{U}$ of scenarios with $2 \le |\overline{\mathcal{U}}| \le n + 1$ such that*

$$r_\varepsilon^*(\mathcal{U}) = r_\varepsilon(x^*, \mathcal{U}) = r_\varepsilon(x^*, \overline{\mathcal{U}}) = r_\varepsilon^*(\overline{\mathcal{U}}).$$

Proof. Let x^* be optimal for (Rec(ε)). The result is trivial if $r_\varepsilon(x^*, \mathcal{U}) = 0$: take any collection of $n + 1$ scenarios. Hence, we may assume $r_\varepsilon(x^*, \mathcal{U}) > 0$, which implies that x^* does not belong to all sets $\mathcal{G}_\varepsilon(\xi)$.

By Lemma 4, $|WC_\varepsilon(x^*, \mathcal{U})| \ge 2$. If $|WC_\varepsilon(x^*, \mathcal{U})| \le n + 1$, then we are done. Otherwise, $|WC_\varepsilon(x^*, \mathcal{U})| > n + 1$, we have by the optimality of x^* and convexity of the functions $d(\cdot, \mathcal{G}_\varepsilon(\xi))$, that

$$0 \in \operatorname{conv}\{\partial d(x^*, \mathcal{G}_\varepsilon(\xi)) : \xi \in WC_\varepsilon(x^*, \mathcal{U})\}$$

By Carathéodory's theorem, $WC_\varepsilon(x^*, \mathcal{U})$ contains a subset $\overline{\mathcal{U}}, 1 \le |\overline{\mathcal{U}}| \le n + 1$ such that $0 \in \operatorname{conv}\left\{\partial d(x^*, \mathcal{G}_\varepsilon(\xi)) : \xi \in \overline{\mathcal{U}}\right\}$. Such $\overline{\mathcal{U}}$ clearly satisfies the conditions stated. □

4.2.2. Problems with linear constraints and polyhedral norms as recovery costs

As in the section before, we assume a finite uncertainty set $\mathcal{U} = \{\xi^1, \ldots, \xi^N\}$. Let us now consider the case that all sets $\mathcal{G}_\varepsilon(\xi^k)$, $k = 1, \ldots, N$ are polyhedral sets. More precisely, we consider problems of type

$$\begin{aligned} \text{P}(\xi) \qquad & \min\ f(x, \xi) := c(\xi)^t x \\ & \text{s.t. } F(x, \xi) := A(\xi)x - b(\xi) \le 0 \\ & x \in \mathcal{X} \end{aligned}$$

with a finite uncertainty set $\mathcal{U} = \{\xi^1, \ldots, \xi^N\}$, linear constraints $A(\xi)x \le b(\xi)$ for every $\xi \in \mathcal{U}$, a linear objective function $c(\xi)^t x$ and a polyhedron $\mathcal{X}$.

Furthermore, let us assume that the distance d is induced by a block norm $\|\cdot\|$, i.e., a norm whose unit ball is a polytope, see Ward, Wendell, and Richard (1985); Witzgall (1964). The most prominent examples for block norms are the Manhattan (d_1) and the maximum (d_∞) norm, which both may be suitable to represent recovery costs: In the case that the recovery costs are obtained by adding single costs of each component, the Manhattan norm is the right choice. The maximum norm may represent the recovery time in the case that a facility has to be moved along each coordinate (or a schedule has to be updated by a separate worker in every component) and the longest time determines the time for the complete update.

We also remark that it is possible to approximate any given norm arbitrarily close by block norms, since the class of block norms is a dense subset of all norms, see Ward et al. (1985). Thus, the restriction to the class of block norms may not be a real restriction in a practical setting.

The goal of this section is to show that under the assumptions above, (Rec(ε)) is a linear program.

We start with some notation. Given a norm $\|\cdot\|$, let

$$B = \{x \in \mathbb{R}^n : \|x\| \le 1\}$$

denote its unit ball. Recall that the unit ball of a block norm $\|\cdot\|$ is a full-dimensional convex polytope which is symmetric with respect to the origin. Since such a polytope has a finite number S of extreme points, we may denote in the following the extreme points of B as

$$\operatorname{Ext}(B) = \{e_i : 1 \le i \le S\}.$$

Since B is symmetric with respect to the origin, $S \in \mathbb{N}$ is always an even number and for any $e_i \in \operatorname{Ext}(B)$ there exists another $e_j \in \operatorname{Ext}(B)$ such that $e_i = -e_j$. Its dual (or polar) norm defined as $\|x\|_0 := \max\{x^t y \colon \|y\| \le 1\}$ has the unit ball

$$B^0 = \{x \in \mathbb{R}^n : x^t y \le 1 \text{ for all } y \in B\}.$$

It is known that B^0 is again a polyhedral norm with extreme points

$$\operatorname{Ext}(B^0) = \{e_i^0 : 1 \le i \le S^0\},$$

where S^0 is the number of facets of B (see, e.g., Rockafellar, 1970).

The following property is crucial for the linear programming formulation of (Rec(ε)). It shows that it is sufficient to consider only the extreme points Ext(B) of either the unit ball B of the block norm, or of the unit ball B^0 of its polar norm in order to compute $\|x\|$ for any point $x \in \mathbb{R}^n$.

Lemma 6 (Ward et al. (1985))**.** *Let $\operatorname{Ext}(B) = \{e_i : 1 \le i \le S\}$ be the extreme points of a block norm $\|\cdot\|$ with unit ball B and let $\operatorname{Ext}(B^0) = \{e_i^0 : 1 \le i \le S^0\}$ be the extreme points of its polar norm with unit ball B^0. Then $\|\cdot\|$ has the following two characterizations:*

$$\|x\| = \min\left\{\sum_{i=1}^{S} \beta_i : x = \sum_{i=1}^{S} \beta_i e_i,\ \ \beta_i \ge 0\ \forall\, i = 1, \ldots, S\right\}$$

and

$$\|x\| = \max_{i=1,\ldots,S^0} x^t e_i^0.$$

Lemma 6 implies that we can compute $\|x-y\|$ for any pair $x, y \in \mathbb{R}^n$ by linear programming. Thus, our assumptions on the sets $\mathcal{G}_\varepsilon(\xi^k)$ and Lemma 6 give rise to the following linear formulations of (Rec(ε)), if $\mathcal{X}$ is a polyhedron:

$$\min \quad r$$

$$\text{s.t. } A(\xi^k)y^k \le b(\xi^k) \qquad \text{for all } k = 1, \dots, N \tag{10}$$

$$c(\xi^k)^t y^k \le \varepsilon \qquad \text{for all } k = 1, \dots, N \tag{11}$$

$$y^k - x = \sum_{i=1}^{S} \beta_i^k e_i \qquad \text{for all } k = 1, \dots, N \tag{12}$$

$$\sum_{i=1}^{S} \beta_i^k \le r \qquad \text{for all } k = 1, \dots, N \tag{13}$$

$$x, y^k \in \qquad \mathcal{X} \text{ for all } k = 1, \dots, N \tag{14}$$

$$r, \beta_i^k \ge 0 \qquad \text{for all } k = 1, \dots, N, \ i = 1, \dots, S \tag{15}$$

Note that constraints (10) and (11) are just the definition of the sets $\mathcal{G}_\varepsilon(\xi^k)$. Furthermore, (12) and (13) together ensure that $\|x - y^k\| \le r$ for all $k = 1, \dots, N$. Hence, the linear program is equivalent to the formulation (9) for a finite set of scenarios each of them having a polyhedron as feasible set and if a block norm is used as distance measure. In this case we have hence shown that (Rec(ε)) can be formulated as a linear program. In order to use the second characterization of block norms in Lemma 6 we replace (12) and (13) by

$$\left(e_i^0\right)^t (y^k - x) \le r \qquad \text{for all } k = 1, \dots, N, \ i = 1, \dots, S^0 \tag{16}$$

to ensure that the value of $\|x - y^k\|$ is correctly computed. We summarize our findings in the following result.

Theorem 3. *Consider an uncertain linear optimization problem* ($\mathrm{P}(\xi), \xi \in \mathcal{U}$) *Let* $\mathcal{U} = \{\xi^k : k = 1, \dots, N\}$ *be a finite set and let d be induced by a block norm. Let* $\mathcal{X} \subseteq \mathbb{R}^n$ *be a polyhedron. Then (Rec(*ε*)) can be solved by linear programming.*

If the number of constraints defining $\mathcal{X}$, *and either the number of extreme points of B or the number of facets of B depend at most polynomially on the dimension n, then (Rec(*ε*)) can be solved in polynomial time.*

We note that block norms may be generalized to the broader class of polyhedral gauges where the symmetry assumption on the unit ball is dropped (see e.g., Nickel & Puerto, 2009). Nevertheless it is readily shown that Lemma 6 applies to polyhedral gauges as well. Hence, it follows that Theorem 3 also holds for distance functions derived from polyhedral gauges.

4.2.3. Problems with hyperplanes as feasible sets

We consider a special case in which (Rec(ε)) can be rewritten as a linear program, even though the distance measure does not need to be derived from a block norm, namely if the sets $\mathcal{G}_\varepsilon(\xi)$ are all hyperplanes or halfspaces. Before we show the resulting linear program for this case, we consider some situations in which this happens:

Example 4. Let d be a distance derived from a norm, and let $\mathcal{X} = \mathbb{R}^n$.

1. For feasibility problems of type

$$\mathrm{P}(a, b) \qquad \min \textit{const}$$
$$\text{s.t. } F(x, (a, b)) := a^t x - b = 0$$

with $\mathcal{U} = \{(a^1, b^1), \dots, (a^N, b^N)\}$, $a^1, \dots, a^N \neq 0$ we obtain $\mathcal{G}_\varepsilon(a^k, b^k) = \{x : a^{k^t} x - b^k = 0\}$ for all $\varepsilon > \textit{const}$.

2. The same holds for problems

$$P(\xi) \quad \min\{f(x, \xi) : x \in \mathcal{F}(\xi)\} \text{ for } \xi \in \mathcal{U}$$

if $\mathcal{F}(\xi)$ is a hyperplane for each $\xi \in \mathcal{U}$ and $\varepsilon > f(x, \xi)$ for all $x \in \mathcal{F}(\xi)$.

3. For unconstrained uncertain linear optimization of the form

$$P(\xi) \quad \min\{c(\xi)^t x : x \in \mathbb{R}^n\}$$

the resulting sets $\mathcal{G}_\varepsilon(\xi^k) = \{x : c(\xi)^t x \le \varepsilon\}$ are halfspaces.

Let us first consider the case of hyperplanes: For $\xi^k = (a^k, b^k)$, let $\mathcal{G}_\varepsilon(\xi) = H_{a^k, b^k} = \{x \in \mathbb{R}^n : a^{k^t} x = b^k\}$ be a hyperplane. Then (Rec(ε)) is given by

$$\min r$$
$$\text{s.t. } d(x, H_{a^k, b^k}) \le r \text{ for all } k = 1, \dots, N$$
$$x \in \mathbb{R}^n, r \in \mathbb{R},$$

Recall the point-to-hyperplane distance (Plastria & Carrizosa, 2001)

$$d(x, H_{a,b}) = \frac{|a^t x - b|}{\|a\|^\circ},$$

where $\|\cdot\|^\circ$ denotes the dual norm to $\|\cdot\|$. As the values of $\|a^k\|^\circ$ can be precomputed and the absolute value linearized, we gain a linear program

$$\min r$$
$$\text{s.t. } -r \le \frac{a^{k^t} x - b}{\|a^k\|^\circ} \le r \text{ for all } k = 1, \dots, N \tag{17}$$
$$x \in \mathbb{R}^n, r \in \mathbb{R}.$$

For halfspaces $\mathcal{G}_\varepsilon(\xi^k) = H^+_{a^k, b^k} = \{x \in \mathbb{R}^n : a^{k^t} x \le b^k\}$ instead of hyperplanes, the distance is given by

$$d(x, H^+_{a,b}) = \frac{|a^t x - b|^+}{\|a\|^\circ},$$

where $|a^t x - b|^+ = \max\{a^t x - b, 0\}$, resulting in the linear program

$$\min r$$
$$\text{s.t. } \frac{a^{k^t} x - b}{\|a^k\|^\circ} \le r \text{ for all } k = 1, \dots, N \tag{18}$$
$$r \ge 0$$
$$x \in \mathbb{R}^n, r \in \mathbb{R}.$$

Theorem 4. *Consider an uncertain optimization problem with finite uncertainty set and sets* $\mathcal{G}_\varepsilon(\xi)$ *that are hyperplanes or halfspaces. Let* $\mathcal{X} = \mathbb{R}^n$ *and let d be derived from a norm* $\|\cdot\|$. *Then (Rec(*ε*)) can be formulated as linear program (see (17) and (18)) and be solved in polynomial time, provided that the dual norm of* $\|\cdot\|$ *can be evaluated in polynomial time.*

5. Reduction approaches

In this section we analyze recoverable-robust solutions for different uncertainty sets $\mathcal{U}$, and hence write Rec($\mathcal{U}$), $\mathbf{f}_\mathcal{U}$ and $\mathbf{r}_\mathcal{U}$ to emphasize the uncertainty set that is considered:

$$\begin{aligned} \text{Rec}(\mathcal{U}) \quad & \text{minimize } (\mathbf{f}_{\mathcal{U}}(y), \mathbf{r}_{\mathcal{U}}(x,y)) \\ & = \left(\sup_{\xi\in\mathcal{U}} f(y(\xi),\xi), \sup_{\xi\in\mathcal{U}} d(x, y(\xi)) \right) \\ & \text{s.t. } F(y(\xi),\xi) \le 0 \quad \text{for all } \xi\in\mathcal{U} \\ & x \in \mathcal{X}, y: \mathcal{U}\to\mathcal{X} \end{aligned}$$

Recall that a solution x is *recoverable-robust with respect to* $\mathcal{U}$ if there exists $y:\mathcal{U}\to\mathcal{X}$ such that (x, y) is Pareto-efficient for Rec($\mathcal{U}$).

The main goal of this section is to reduce the set $\mathcal{U}$ to a smaller (maybe even finite) set $\mathcal{U}'\subseteq\mathcal{U}$, such that the set of recovery-robust solutions does not change. This is the case if we can extend any feasible solution (x, y') for Rec($\mathcal{U}'$) to a feasible solution (x, y) for Rec($\mathcal{U}$) without changing the objective function values.

Lemma 7. *Let $\mathcal{U}'\subseteq\mathcal{U}$. If for all feasible solutions (x, y') of Rec($\mathcal{U}'$) there exists $y:\mathcal{U}\to\mathcal{X}$ such that*

- *(x, y) is feasible for Rec($\mathcal{U}$), i.e., $F(y(\xi), \xi) \le 0$ for all $\xi\in\mathcal{U}$, and*
- *$\mathbf{f}_{\mathcal{U}}(y) = \mathbf{f}_{\mathcal{U}'}(y')$ and $\mathbf{r}_{\mathcal{U}}(x,y) = \mathbf{r}_{\mathcal{U}'}(x,y')$*

then Rec($\mathcal{U}$) and Rec($\mathcal{U}'$) have the same recoverable-robust solutions.

Proof. Let (x, y) be feasible for Rec($\mathcal{U}$). Define

$$y_{|\mathcal{U}'}:\mathcal{U}'\to\mathcal{X} \quad \text{through } y_{|\mathcal{U}'}(\xi) := y(\xi) \quad \text{for all } \xi\in\mathcal{U}'$$

Then (x, y') is feasible for Rec($\mathcal{U}'$) and $\mathbf{f}_{\mathcal{U}'}(y') \le \mathbf{f}_{\mathcal{U}}(y)$, $\mathbf{r}_{\mathcal{U}'}(x,y') \le \mathbf{r}_{\mathcal{U}}(x,y)$. Together with the assumption of this lemma Pareto optimality follows since a solution can be improved by switching between Rec($\mathcal{U}$) and Rec($\mathcal{U}'$):

- Let x be recoverable-robust w.r.t $\mathcal{U}$. Then there exists $y:\mathcal{U}\to\mathcal{X}$ such that (x, y) is Pareto efficient for Rec($\mathcal{U}$). Define $y' := y_{|\mathcal{U}'}$. Then (x, y') is Pareto-efficient for Rec($\mathcal{U}'$): Assume that $(\tilde{x},\tilde{y}')$ dominates (x, y'). Due to the assumption of this lemma there exists $(\tilde{x},\tilde{y})$ which is feasible for Rec($\mathcal{U}$) and $\mathbf{f}_{\mathcal{U}}(\tilde{y}) = \mathbf{f}_{\mathcal{U}'}(\tilde{y}')$ and $\mathbf{r}_{\mathcal{U}}(\tilde{x},\tilde{y}) = \mathbf{r}_{\mathcal{U}'}(\tilde{x},\tilde{y}')$, i.e., $(\tilde{x},\tilde{y})$ then dominates (x, y), a contradiction.
- Let x be recoverable-robust w.r.t $\mathcal{U}'$. Then there exists $y':\mathcal{U}'\to\mathcal{X}$ such that (x, y') is Pareto-efficient for Rec($\mathcal{U}'$). Choose y according to the assumption of this lemma. Then (x, y) is Pareto-efficient for Rec($\mathcal{U}$): Assume that $(\tilde{x},\tilde{y})$ dominates (x, y). Then $(\tilde{x},\tilde{y}_{|\mathcal{U}'})$ is feasible for Rec($\mathcal{U}'$) and $\mathbf{f}_{\mathcal{U}'}(\tilde{y}_{|\mathcal{U}'}) \le \mathbf{f}_{\mathcal{U}}(\tilde{y})$ and $\mathbf{r}_{\mathcal{U}'}(\tilde{x},\tilde{y}_{|\mathcal{U}'}) \le \mathbf{r}_{\mathcal{U}}(\tilde{x},\tilde{y})$, i.e., $(\tilde{x},\tilde{y}_{|\mathcal{U}'})$ then dominates (x, y'), a contradiction. □

We now use Lemma 7 to reduce the set of scenarios $\mathcal{U}$. Our first result is similar to the reduction rules for set covering problems (Toregas, Swain, ReVelle, & Bergman, 1971).

Lemma 8. *If P(ξ^2) is a relaxation of P(ξ^1) for two scenarios $\xi^1, \xi^2 \in \mathcal{U}$, then Rec($\mathcal{U}$) and Rec($\mathcal{U}\setminus\{\xi^2\}$) have the same recoverable robust solutions, i.e., scenario ξ^2 may be ignored.*

Proof. We check the condition of Lemma 7: Let (x, y') be feasible for Rec($\mathcal{U}\setminus\{\xi^2\}$). Define

$$y:\mathcal{U}\to\mathcal{X} \quad \text{through } y(\xi) := \begin{cases} y'(\xi) & \text{if } \xi\in\mathcal{U}\setminus\{\xi^2\} \\ y'(\xi^1) & \text{if } \xi = \xi^2 \end{cases}$$

Then (x, y) is feasible since $F(y(\xi), \xi) \le 0$ for all $\xi\in\mathcal{U}\setminus\{\xi^2\}$ and $F(y(\xi^2),\xi^2) = F(y(\xi^1),\xi^2) \le 0$ since $F(y(\xi^1), \xi^1) \le 0$ and P(ξ^2) is a relaxation of P(ξ^1). Furthermore, $f(y(\xi^2),\xi^2) = f(y(\xi^1),\xi^2) \le f(y(\xi^1),\xi^1)$ implies

$$\begin{aligned} \mathbf{f}_{\mathcal{U}}(y) = \sup_{\xi\in\mathcal{U}} f(y(\xi),\xi) &= \sup_{\xi\in\mathcal{U}\setminus\{\xi^2\}} f(y(\xi),\xi) \\ &= \sup_{\xi\in\mathcal{U}\setminus\{\xi^2\}} f(y'(\xi),\xi) = \mathbf{f}_{\mathcal{U}\setminus\{\xi^2\}}(y'). \end{aligned}$$

Finally, $y(\xi^1) = y(\xi^2)$, hence

$$\begin{aligned} \mathbf{r}_{\mathcal{U}}(x,y) = \sup_{\xi\in\mathcal{U}} d(x,y(\xi)) &= \sup_{\xi\in\mathcal{U}\setminus\{\xi^2\}} d(x,y(\xi)) \\ &= \sup_{\xi\in\mathcal{U}\setminus\{\xi^2\}} d(x,y'(\xi)) = \mathbf{r}_{\mathcal{U}\setminus\{\xi^2\}}(x,y'). \end{aligned}$$

□

Note that depending on the definition of the optimization problem and the uncertainty set $\mathcal{U}$, often large classes of scenarios may be dropped. This is in particular the case if the sets $\mathcal{F}(\xi)$ are nested.

In the following we are interested in identifying a kind of *core set*$\mathcal{U}'\subseteq\mathcal{U}$ containing a *finite* number of scenarios which are sufficient to consider in order to solve the recoverable-robust counterpart. More precisely, we look for a finite set $\mathcal{U}'$ such that Rec($\mathcal{U}$) and Rec($\mathcal{U}'$) have the same recoverable-robust solutions.

In the following we consider a polytope $\mathcal{U}$ with a finite number of extreme points $\xi^1, \ldots, \xi^N$, i.e., let

$$\mathcal{U} = conv(\mathcal{U}') \quad \text{where} \quad \mathcal{U}' = \{\xi^1, \ldots, \xi^N\}.$$

Then many robustness concepts have (under mild conditions) the following property: Instead of investigating *all* $\xi\in\mathcal{U}$, it is enough to investigate the extreme points $\xi^1, \ldots, \xi^N$ of $\mathcal{U}$. For example, for the strictly robust counterpart RC($\mathcal{U}$) of an uncertain optimization problem $(P(\xi), \xi\in\mathcal{U} = conv\{\xi^1, \ldots, \xi^N\})$, RC($\mathcal{U}$) is equivalent to RC($\{\xi^1, \ldots, \xi^N\}$), if $F(x, \cdot)$ is convex for all $x\in\mathcal{X}$.

Unfortunately, a similar result for the recoverable-robust counterpart does not hold. This means that the set of Pareto efficient solutions of Rec($\mathcal{U}'$) does in general not coincide with the set of Pareto efficient solutions of Rec($\mathcal{U}$) with respect to the larger set $\mathcal{U} = conv(\mathcal{U}')$ as demonstrated in the following example.

Example 5. Consider the following uncertain optimization problem:

$$\begin{aligned} P(a_1, a_2, b) \qquad & \min\ f(x_1, x_2) = const \\ & \text{s.t. } a_1x_1 + a_2x_2 - b = 0 \\ & x_1, x_2 \in \mathbb{R}, \end{aligned}$$

where

$$\mathcal{U} = conv(\mathcal{U}') \text{ with } \mathcal{U}' = \{(1,0,0), (0,1,0), (1,1,2)\}.$$

Let the recovery distance be the Euclidean distance. Then $x^* = (2-\sqrt{2}, 2-\sqrt{2})$, the midpoint of the incircle of the triangle that is given by the intersections of the respective feasible solutions, is a Pareto efficient solution of Rec($\mathcal{U}'$), as it is the unique minimizer of the recovery distance (see Fig. 3(a)).

On the other hand, this solution is not Pareto efficient when the convex hull of $\mathcal{U}'$ is taken into consideration. Indeed, by elementary geometry, one finds that

$$r(x^*, \mathcal{U}) = \sqrt{2}\cdot(2-\sqrt{2}) \approx 0.828,$$

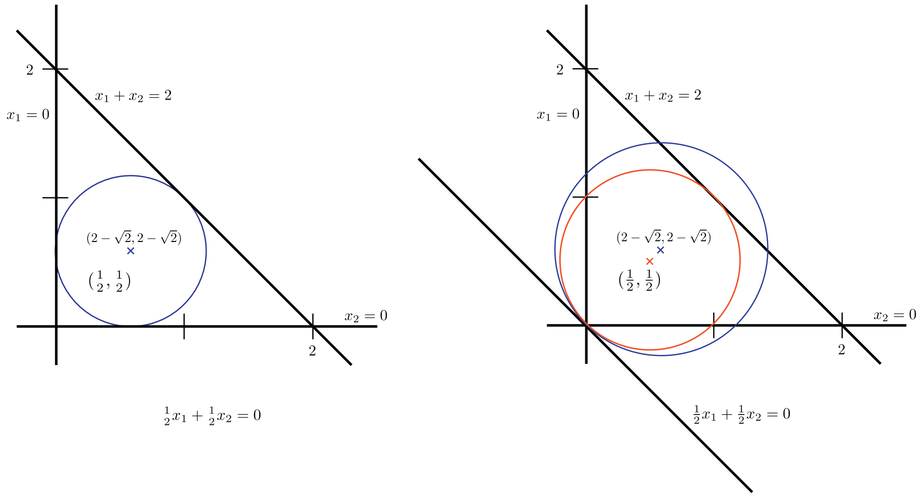
$$r(\bar{x}, \mathcal{U}) = \frac{1}{\sqrt{2}} \approx 0.707,$$

where $\bar{x} = (\frac{1}{2}, \frac{1}{2})$ (see Fig. 3(b)). Therefore, solving Rec($\mathcal{U}'$) does not give the set of Pareto efficient solutions for Rec($\mathcal{U}$) .

However, assuming more problem structure, we can give the following result.

Theorem 5. *Consider an uncertain optimization problem with uncertainty set $\mathcal{U} = conv(\mathcal{U}')$ with $\mathcal{U}' := \{\xi^1, \ldots, \xi^N\}$. Let F consist of m constraints with $F_i : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}$, $i = 1, \ldots, m$ and $f : \mathbb{R}^n \times \mathcal{U} \to \mathbb{R}$ be jointly quasiconvex in the arguments (y, ξ). Let $d(x, \cdot)$ be quasiconvex. Let $\mathcal{X}$ be convex.*

Then Rec($\mathcal{U}$) and Rec($\mathcal{U}'$) have the same set of recoverable-robust solutions.


2
$x_1 = 0$
$x_1 + x_2 = 2$
$(2-\sqrt{2}, 2-\sqrt{2})$
$(\frac{1}{2}, \frac{1}{2})$
$x_2 = 0$
2
$\frac{1}{2}x_1 + \frac{1}{2}x_2 = 0$
2
$x_1 = 0$
$x_1 + x_2 = 2$
$(2-\sqrt{2}, 2-\sqrt{2})$
$(\frac{1}{2}, \frac{1}{2})$
$x_2 = 0$
2
$\frac{1}{2}x_1 + \frac{1}{2}x_2 = 0$

(a) Optimal solution w.r.t. $\mathcal{U}'$.

(b) Optimal solution w.r.t. $\mathcal{U}$.

Fig. 3. Rec($\mathcal{U}'$) and Rec($\mathcal{U}$) may have different optimal solutions.

Proof. Let (x, y') be feasible for Rec($\mathcal{U}'$). We first define $y : \mathcal{U} \to \mathcal{X}$.

Let $\xi \in \mathcal{U}$. Then there exist $\lambda_i, i = 1, \ldots, N$ such that $\xi = \sum_{i=1}^N \lambda_i \xi^i$ with $\sum_{i=1}^N \lambda_i = 1$ and $\lambda_i \geq 0$. We set $y(\xi) := \sum_{i=1}^N \lambda_i y'(\xi^i)$. Note that this implies $y(\xi^i) = y'(\xi^i)$ for all $i = 1, \ldots, N$. We now check the conditions of Lemma 7.

For every constraint $k = 1, \ldots, m$ the joint quasiconvexity implies that

$$F_k(y(\xi), \xi) = F_k\left(\sum_{i=1}^N \lambda_i y(\xi^i), \sum_{i=1}^N \lambda_i \xi^i\right)$$
$$\leq \max_{i=1,\ldots,N} F_k(y(\xi^i), \xi^i) \leq 0 \ \forall k = 1, \ldots, m,$$

where the last inequality holds since $y(\xi^i) = y'(\xi^i)$ and (x, y') is feasible for Rec($\mathcal{U}'$). We hence have that (x, y) is feasible for Rec($\mathcal{U}$).

Analogously, joint quasiconvexity of f implies $f(y(\xi), \xi) \leq \max_{i=1,\ldots,N} f(y(\xi^i), \xi^i)$ for all $\xi \in \mathcal{U}$, hence

$$\mathbf{f}_{\mathcal{U}}(y) = \sup_{\xi \in \mathcal{U}} f(y(\xi), \xi) = \max_{\xi \in \mathcal{U}'} f(y(\xi), \xi)$$
$$= \max_{\xi \in \mathcal{U}'} f(y'(\xi), \xi) = \mathbf{f}_{\mathcal{U}'}(y').$$

Finally, for the recovery distance d we assumed quasiconvexity in its second argument which implies $d(x, y(\xi)) \leq \max_{i=1,\ldots,N} d(x, y(\xi^i))$, hence

$$\mathbf{r}_{\mathcal{U}}(x, y) = \sup_{\xi \in \mathcal{U}} d(x, y(\xi)) = \max_{\xi \in \mathcal{U}'} d(x, y(\xi))$$
$$= \max_{\xi \in \mathcal{U}'} d(x, y'(\xi)) = \mathbf{r}_{\mathcal{U}'}(x, y').$$

□

An important particular case of Theorem 5 is the case in which

$$F(x, \xi) = G(x) - b(\xi)$$

for a convex G and concave b (i.e., the uncertainty is in the right-hand side), since F is then jointly quasiconvex in (x, ξ).

Corollary 1. *Let* $(\mathrm{P}(\xi), \xi \in \mathcal{U})$ *be an uncertain optimization problem with uncertainty set* $\mathcal{U} = conv(\mathcal{U}')$ *with* $\mathcal{U}' := \{\xi^1, \ldots, \xi^N\}$. *Let* $F(x, \xi) = G(x) - b(\xi)$ *with a convex function* $G : \mathbb{R}^n \to \mathbb{R}^m$ *and a concave function* $b(\xi) : \mathbb{R}^M \to \mathbb{R}^m$. *Let f(x, ξ) be jointly quasiconvex,* $\mathcal{X}$ *be convex, and let d(x, ·) be quasiconvex. Then Rec(*$\mathcal{U}$*) and Rec(*$\mathcal{U}'$*) have the same recoverable-robust solutions.*

We remark that G must not depend on the scenario ξ. Example 5 shows that Corollary 1 is not even true for a linear function $F(x, \xi) = A(\xi)x - b(\xi)$: If the matrix A is dependent on ξ, we cannot conclude that Rec($\mathcal{U}$) and Rec($\mathcal{U}'$) have the same recoverable-robust solutions.

Note that Corollary 1 applies in particular for the special case where $b(\xi) = \xi$, i.e., for uncertain convex optimization problems of the type

$$\mathrm{P}(b) \ \min_{x \in \mathbb{R}^n} \{f(x) : G(x) \leq b\}. \tag{19}$$

In particular we know for P(b) that the center with respect to some finite set $\mathcal{U}'$ solves the uncertain problem with respect to $\mathcal{U} = conv(\mathcal{U}')$.

This means we can use the finite set $\mathcal{U}'$ instead of $\mathcal{U}$ when solving (Rec) if the conditions of the previous theorem apply. This is summarized next.

Corollary 2. *Let* $(\mathrm{P}(\xi), \xi \in \mathcal{U})$ *be an uncertain optimization problem with uncertainty set* $\mathcal{U} = conv(\mathcal{U}')$ *with* $\mathcal{U}' := \{\xi^1, \ldots, \xi^N\}$ *and with constraints* $F(x, \xi) = G(x) - b(\xi)$ *with a convex function* $G : \mathbb{R}^n \to \mathbb{R}^m$ *and a concave function* $b(\xi) : \mathbb{R}^M \to \mathbb{R}^m$. *Let* $\mathcal{X} \subseteq \mathbb{R}^n$ *be convex, let f be jointly convex, and let d(x, ·) be convex. Then (Rec) can be formulated as the following convex biobjective program:*

$$\begin{array}{llll} \min & (r, z) & & \\ \text{s.t.} & G(y^k) & \leq b(\xi^k) & \text{for all } k = 1, \ldots, N \\ & d(x, y^k) & \leq r & \text{for all } k = 1, \ldots, N \\ & f(y^k, \xi^k) & \leq z & \text{for all } k = 1, \ldots, N \\ & x, y^k & \in \mathcal{X} & \text{for all } k = 1, \ldots, N \\ & r, z & \in \mathbb{R} & \end{array} \tag{20}$$

Combining this corollary with Theorem 3 from Section 4.2.2, we obtain the following result: The recoverable-robust counterpart of an optimization problem with convex uncertainty which is only in its right-hand side and with polyhedral uncertainty set can be formulated as a linear program if a block norm is used to measure the recovery costs. In particular, the recoverable-robust counterpart of such a linear program under polyhedral uncertainty sets and block norms as distance functions remains a linear program.

Theorem 6. *Let* $(\mathrm{P}(\xi), \xi \in \mathcal{U})$ *be an uncertain linear program with concave uncertainty only in the right-hand side, and* $\mathcal{U} = conv(\mathcal{U}')$ *with* $\mathcal{U}' := \{\xi^1, \ldots, \xi^N\}$. *Let d be derived from a block norm. Then, (Rec) can be formulated as a linear biobjective program.*

If the terms defining $\mathcal{X}$ and either the number of extreme points or the number of facets of the unit ball of the block norm depend at most polynomially on the dimension n, then the problem (Rec(ε)) be solved in polynomial time.

Proof. According to Theorem 1 we can replace $\mathcal{U}$ by the finite set $\mathcal{U}'$ in the recoverable-robust counterpart, i.e., we consider Rec($\mathcal{U}'$) instead of Rec($\mathcal{U}$). We are hence left with a problem for which the assumptions of Theorem 3 are satisfied yielding a formulation as linear program. □

Note that many practical applications satisfy the conditions of Theorem 6. Among these are scheduling and timetabling problems where the uncertainty is the length of the single tasks to be completed and hence in the common linear formulations in the right-hand side. We refer to Goerigk and Schöbel (2010) for applications in timetabling, to Herroelen and Leus (2005) for project scheduling, to Erera et al. (2009) for container repositioning, and to Bouman, van den Akker, and Hoogeveen (2011) for knapsack problems.

6. Numerical experiments

In the following, we consider two computational experiments: The first experiment is concerned with computation times on randomly generated portfolio optimization instances. We analyze the difference between our scalarization (Rec(ε)) and the "classic" scalarization (Rec$^{\text{class}}(\delta)$) to calculate the Pareto front. In the second experiment, we use real-world stock exchange data and analyze the performance of the biobjective model.

Note that both recovery distance and worst-case performance are relevant quality measures in portfolio optimization. On the one hand, a good worst-case performance ensures that the solution will perform well over all scenarios. On the other hand, however, the small recovery distance adds an element of stability to the optimization, as it can be used to avoid solutions where only few stocks are bought. This stability then translates to additional robustness over time.

6.1. Experiment 1: computation times

6.1.1. Problem setting

We consider a portfolio problem of the form

$$
\begin{aligned}
\max\ & \sum_{i=1}^{n} p_i x_i \\
\text{s.t.}\ & \sum_{i=1}^{n} x_i = 1 \\
& x \geq 0
\end{aligned}
$$

where variable x_i denotes the amount of investment in opportunity $i \in \{1, \dots, n\}$ with profit p_i. We assume that profits are uncertain and stem from a finite uncertainty set $\mathcal{U} = \{p^1, \dots, p^N\} \subseteq \mathbb{R}^n_+$. The biobjective recoverable-robust model we would like to solve is the following:

$$
\begin{aligned}
& (\max\ z,\ \min\ d) \\
\text{s.t.}\ & z \leq \sum_{i=1}^{n} p_i^k x_i^k && \forall k = 1, \dots, N \\
& \sum_{i=1}^{n} x_i = 1 \\
& \sum_{i=1}^{n} x_i^k = 1 && k = 1, \dots, N \\
& \sum_{i=1}^{n} (x_i - x_i^k)^2 \leq d && \forall k = 1, \dots, N \\
& x, x^k \geq 0
\end{aligned}
$$

In this setting, we would like to fix some choice of investment x now, but can modify it, once the scenario becomes known. Our aim is to maximize the resulting worst-case profit, and also to minimize the modifications to our investment, which we measure by using the Euclidean distance.

We compare the two ε-constraint approaches, where either a fixed budget on d is given (Rec$^{\text{class}}(\delta)$), or a budget on z is given (Rec(ε)).

Moreover, we consider the following iterative *projection method* as another solution approach to (Rec(ε)) It is based on the method of alternating projections. Say we have some candidate solution x available. For every scenario k, we want to find a solution x^k that is as close to x as possible, and also respects a desired profit bound P. The resulting problems are independent for every k. For a fixed k, it can be formulated as the following quadratic program:

$$
\begin{aligned}
\min\ & \sum_{i=1}^{n} (x_i^k - x_i)^2 \\
\text{s.t.}\ & \sum_{i=1}^{n} x_i^k = 1 \\
& \sum_{i=1}^{n} p_i^k x_i^k \geq P \\
& x^k \geq 0
\end{aligned}
$$

Having calculated all points x^k, we then proceed to find a new solution x' that is as close to all points x^k as possible:

$$
\begin{aligned}
\min\ & d \\
\text{s.t.}\ & \sum_{i=1}^{n} x'_i = 1 \\
& \sum_{i=1}^{n} (x'_i - x_i^k)^2 \leq d && \forall k = 1, \dots, N \\
& x' \geq 0
\end{aligned}
$$

We then repeat the calculation of closest points, until the change in objective value is sufficiently small. In this setting, the projection method is known to converge to an optimal solution (see, e.g., Dattorro, 2010; Goerigk, 2012)

6.1.2. Instances and computational setting

We consider instances with $n = 5, 10, 15, 20, 25, 30$ and $N = 5, 10, 15, 20, 25, 30$, where we generate 100 instances for each setting of n and N (i.e., a total of $6 \cdot 6 \cdot 100 = 3600$ instances were generated). An instance is generated by sampling uniformly randomly values for p_i^k in the range $\{1, \dots, 100\}$.

For each instance, we first calculate the two lexicographic solutions with respect to recovery distance and profit. Then the following problems were solved:

- We solve the classic scalarization, (Rec$^{\text{class}}(\delta)$), i.e., (Rec) with bounds on the recovery distance, where the bounds are calculated by choosing 50 equidistant points within the relevant region given by the lexicographic solutions. This approach is denoted as Rec-P.
- For solving the new scalarization, i.e., (Rec(ε)), we used three different approaches:
    - Using also 50 equidistant bounds on the profit, we solve recoverable-robust problems (Rec(ε)) directly. This approach is denoted as Rec-D.

Table 1
Average computation times in *s* to calculate Pareto solutions.

n	N	Rec-P	Rec-D	Rec-It	Rec-M
5	5	**0.29**	0.32	1.70	0.48
	10	**0.48**	0.56	2.56	0.77
	15	**0.74**	0.91	3.43	1.16
	20	**0.99**	1.15	3.78	1.40
	25	**1.26**	1.49	4.14	1.75
	30	1.55	1.86	5.30	2.18
10	5	**0.57**	0.62	3.31	0.74
	10	**1.45**	1.53	6.22	1.67
	15	2.70	**2.59**	8.60	2.79
	20	4.42	**4.11**	13.15	4.33
	25	**3.70**	4.12	17.95	4.99
	30	**4.47**	5.04	21.36	6.38
15	5	**0.85**	0.96	5.08	1.04
	10	2.85	2.97	8.62	**2.84**
	15	5.46	5.13	14.82	**4.94**
	20	10.85	9.16	25.65	**8.80**
	25	18.08	14.56	32.12	**13.31**
	30	**10.37**	20.83	46.30	19.07
20	5	**1.19**	1.25	6.74	1.33
	10	4.86	5.08	13.60	**4.50**
	15	11.23	10.03	25.10	**8.91**
	20	20.48	13.22	34.78	**12.27**
	25	30.02	22.81	49.34	**19.98**
	30	44.38	36.88	65.80	**31.45**
25	5	1.57	**1.51**	8.08	1.59
	10	5.06	**4.22**	19.55	4.23
	15	10.58	8.62	29.81	**8.35**
	20	19.04	15.10	46.93	**14.19**
	25	35.82	28.18	75.60	**26.09**
	30	53.97	42.80	102.47	**38.49**
30	5	2.02	**1.83**	9.77	1.84
	10	6.27	**4.98**	25.59	5.16
	15	13.44	**10.29**	45.68	10.32
	20	24.04	**18.31**	71.05	18.44
	25	39.49	29.53	101.90	**28.90**
	30	68.43	51.67	145.12	**47.77**

- In the same setting as for Rec-D, we use the iterative projection algorithm instead of solving the recovery problem directly with Cplex. This is denoted as Rec-It.
- Finally, as preliminary experiments showed that Rec-It is especially fast if the bound on the profit P is large, we used a mixed approach that uses Rec-D for the 2/3 smallest bounds on P, and Rec-It for the 1/3 largest bounds on P. This is denoted as Rec-M.

We used Cplex v.12.6 to solve the resulting quadratic programs. The experiments were conducted on a computer with a 16-core Intel Xeon E5-2670 processor, running at 2.60 gigahertz with 20 megabytes cache, and Ubuntu 12.04. Processes were pinned to one core.

6.1.3. Results

We show the average computation times for the biobjective portfolio problem in Table 1.

The best average computation time per row is printed in bold. Note that Rec-It requires higher computation times than any other approach; however, in combination with Rec-D (i.e., Rec-M), it is highly competitive. While Rec-P performs well for smaller instances, Rec-D and Rec-M perform best for larger instances.

There are some surprises in Table 1, which are not due to outliers. For Rec-P and $n = 10$, one can see that solving $N = 20$ takes longer than solving $N = 25$. The same holds for $n = 15$, $N = 25$ and $N = 30$. Also, for $N = 15$, we find that Rec-P is faster for $n = 25$ than for $n = 20$ (the same holds for Rec-D). This behavior disappears for large n and N.

Summarizing, our experimental results show that switching perspective from the classic recoverable-robust approach ($\text{Rec}^{\text{class}}(\delta)$) that maximizes the profit subject to some fixed recovery distance to the ($\text{Rec}(\varepsilon)$) approach we suggest, in which the distance is minimized subject to some bound on the profit, results in improved computation times. These computation times are further improved by applying methods from location theory, that can allow the ($\text{Rec}(\varepsilon)$) version to be solved more efficiently.

6.2. Experiment 2: performance on real-world instance

6.2.1. Setting

In this experiment we analyze the performance of the biobjective model on a real-world portfolio instance. We use data of the S&P 100 stock market index consisting of 100 companies that are amongst the largest and most established at the US market exchange. Our dataset consists of daily values of all stocks for the time period 1/4/2010 – 31/12/2015. These 2100 days are divided into 70 periods of 30 days each.

As a starting portfolio, a unit of wealth is equally distributed over all available stock options. After each 30 day period, the portfolio can be changed. When the portfolio is recomputed, we use the last 30 days as one scenario each. Using historical data as scenarios in such a way is considered a simple yet reasonable way to generate meaningful scenarios, see Guastaroba, Mansini, and Speranza (2009).We remark that better scenario sets may be generated by forecasting methods.

As recovery distance, we use the median with respect to the Manhattan norm, and also include the current portfolio in the computation of the distance. We test different weight values $w \in [0, 1]$ between the two objectives from (Rec), namely recovery distance $\mathbf{r}(x, y)$ (in the portfolio case $\min d$) and worst-case performance $\mathbf{f}(y)$ (in the portfolio case $\max z$).

Note that for the special case $w = 0$, the equidistributed starting solution is kept over all periods (i.e., it is an index tracker fund). For the special case $w = 1$, we use a lexicographic post-optimization to ensure that we find an efficient solution (i.e., each scenario is solved separately, and then the median of all solutions with additional portfolio value constraint is computed).

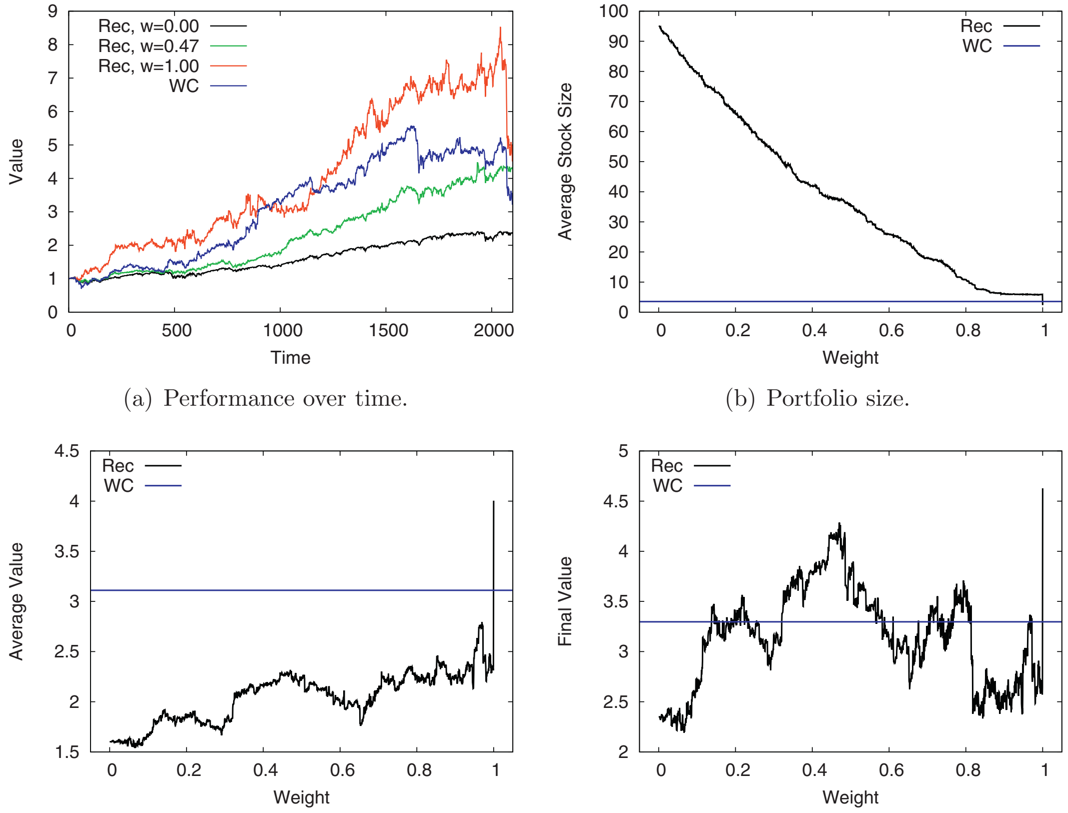
Additionally, we compute a worst-case solution on the same scenario set for comparison; i.e., a single portfolio that maximizes the worst-case performance over the observations of the past 30 days. We denote this solution as “WC” in the following. Note that WC protects only against these past scenarios, but future developments may be different. Furthermore, the worst-case solution may not be “robust” in the sense that it spreads the portfolio over many options, but it may instead use few options that seem to perform well in the worst case.

The purpose of this experiment is to compare the performance and qualitative properties of the approach (Rec) for various objective function weights and also with the worst-case approach. Our aim is not to solve the portfolio problem more efficiently than in the current literature, for which more advanced scenario generation procedures and optimization models can be found in the literature, e.g., Prigent (2007).

6.2.2. Results

We present the key findings in Fig. 4.

The average value of the portfolio over the 2100 day period for different weights w is shown in Fig. 4(c). The horizontal blue line indicates the average value of WC. While the average portfolio value fluctuates over w, there is a slight trend that higher weights (i.e., models where the worst-case performance is weighted higher) perform better on average. For $w = 1$, the solution shows very different performance than for any other value of w. Also, note that nearly all approaches perform better than the simple index tracking solution $w = 0$.



(a) Performance over time.

(b) Portfolio size.

(c) Average wealth.

(d) Final wealth.

Fig. 4. Results of Experiment 2.

The average value is complemented by Fig. 4(d), which shows the value of the portfolio at the end of the 2100 day period. It can be seen that even though some weight values did not perform well on average, they did give a high value in the end, which is often better than WC.

This behavior can be explained by Fig. 4(a) and (b), where the average size of the portfolio (i.e., the average number of stocks i for which $x_i > 0$) and the portfolio value over time for a sample of weight values is shown, respectively.

As can be seen in Fig. 4(b), the weight w and the size of the portfolio are clearly correlated. The maximum average portfolio size is 95, as not all stocks options that are included in this instance were already part of the S&P index at the starting time of the considered time period. The more important the worst-case performance becomes, the less stock options are being used. This gives the potential of high gains, but at the same time, also makes the solution more vulnerable. In this sense, also WC is not “robust”.

In Fig. 4(b), the effect of this vulnerability can be seen shortly before the end of the 2100 day period. Both the solution for $w = 1$ and WC decrease dramatically in value, as one of the few stock options that were being used decreased in value. This did not happen for $w = 0.47$, which spread the wealth over a more diverse set of options. As the uncertainty set cannot make accurate future predictions, this adds another layer of security to the portfolio. The solution with respect to $w = 0$, for example, shows little fluctuation, but is in this sense too conservative to increase in value as the other approaches.

Overall, we find that using both recovery distance and worst-case performance in a biobjective setting can help improving the portfolio value in this real-world example, in particular, the resulting portfolios performed better as the equidistributed starting solution which corresponds to an index tracker fund. Both WC and the lexicographic approach that solves each scenario separately, and then takes the median over all solutions, turn out to use only few of the available stock options. If a scenario occurs that is not part of the uncertainty set (as it is likely in this setting), such a restricted choice can lead to a bad performance overall. By using a weighted sum between both objectives, a good compromise can be found between over-specialization and over-spreading of the portfolio.

7. Summary and conclusion

In this paper, we introduced a location-analysis based point of view to the problem of finding recoverable-robust solutions to uncertain optimization problems. Table 2 summarizes the results we obtained.

The following variation of (Rec) should be mentioned: In many cases it might not be appropriate to just look at the worst-case objective function of the recovered solutions, because there might be one very bad scenario which is the only relevant one. Pareto efficient solutions would hence neglect the objective function values of all other scenarios.

This might lead to another goal, namely to be as close as possible to an optimal solution in *all* scenarios instead of only looking at a few scenarios which will be very bad anyway. This leads to the following problem in which we bound the difference between the objective value of the recovered solution and the best possible objective function value in the worst case:

$$(\widehat{\text{Rec}}) \qquad \text{minimize } \left(\hat{\mathbf{f}}(y), \mathbf{r}(x, y)\right)$$

$$= \left(\sup_{\xi \in \mathcal{U}} f(y(\xi), \xi) - f^*(\xi), \sup_{\xi \in \mathcal{U}} d(x, y(\xi))\right)$$

Table 2
Summary of properties of (Rec) and (Rec(ε)) depending on the optimization problem P(ξ), the uncertainty set $\mathcal{U}$, the type of uncertainty, and the recovery costs.

Uncertainty set $\mathcal{U}$	Constraints $F(\cdot,\xi)$	Uncertainty $F(x,\cdot)$	Rec. costs d	Deterministic constraints $\mathcal{X}$	Results
Finite	Quasiconvex	Arbitrary	Convex	Convex and closed $\mathcal{X}=\mathbf{R}^n$	- (Rec(ε)) convex problem (Lemma 5) - Reduction to (Rec($\bar{\mathcal{U}}$)) for smaller sets $\bar{\mathcal{U}}$ (Theorem 2)
Finite	Linear	Arbitrary	Block norm	Polyhedron	- (Rec(ε)) linear problem (Theorem 3)
Polyhedron	Jointly quasiconvex		Convex	Closed	- Pareto solution w.r.t. extreme points of $\mathcal{U}$ is Pareto (Theorem 5)
Polyhedron	Convex	Quasiconvex, right-hand side	Convex	Closed	- solution w.r.t extreme points of $\mathcal{U}$ is Pareto (Corollary 1)
				Convex and closed	- (Rec(ε)) convex problem (Corollary 2)
Polyhedron	Linear	Quasiconvex, right-hand side	Block norm	Polyhedron	- (Rec(ε)) linear problem (Theorem 6)

$$
\begin{aligned}
\text{s.t. } & F(y(\xi),\xi)\le 0 \quad \text{for all } \xi\in\mathcal{U}\\
& x\in\mathcal{X},\, y:\mathcal{U}\to\mathcal{X}
\end{aligned}
$$

The new objective function $\hat{\mathbf{f}}$ in $(\widehat{\mathrm{Rec}})$ can be interpreted as a minmax-regret approach as described in Kouvelis and Yu (1997). Again, we can look at the scalarizations of this problem. Instead of (Rec(ε))we receive

$$
\begin{aligned}
(\widehat{\mathrm{Rec}}(\varepsilon)) \qquad & \text{minimize } \sup_{\xi\in\mathcal{U}} d(x,y(\xi))\\
\text{s.t. } & f(y(\xi),\xi)-f^*(\xi)\le\varepsilon \quad \text{for all } \xi\in\mathcal{U}\\
& F(y(\xi),\xi)\le 0 \quad \text{for all } \xi\in\mathcal{U}\\
& x\in\mathcal{X},\, y:\mathcal{U}\to\mathcal{X}
\end{aligned}
$$

In case that $f^*(\xi)$ is known for all $\xi\in\mathcal{U}$, $(\widehat{\mathrm{Rec}}(\varepsilon))$ admits similar properties as (Rec(ε)).

Note that the lexicographic solution of $(\widehat{\mathrm{Rec}}(\varepsilon))$ with respect to $(\hat{\mathbf{f}},\mathbf{r})$ requires to find optimal solutions for each scenario $\xi\in\mathcal{U}$ which can be reached with minimal recovery costs. It can be found by solving $(\widehat{\mathrm{Rec}}(0))$. This is exactly the robustness approach *recovery-to-optimality* which has been described in Goerigk and Schöbel (2014), see Goerigk and Schöbel (2010, 2011) for scenario-based approaches for its solution. On the other hand, the lexicographic solution of $(\widehat{\mathrm{Rec}}(\varepsilon))$ with respect to $(\mathbf{r},\hat{\mathbf{f}})$ is related to minmax regret robustness.

Ongoing research includes the analysis of other special cases of (Rec) as well as its application to other types of problems e.g. from traffic planning or evacuation. We also work on generalizations to multi-objective uncertain optimization problems as already done for several minmax robustness concepts (Ehrgott, Ide, & Schöbel, 2014).

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.

An, L. T. H., & Tao, P. D. (2005). The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. *Annals of Operations Research, 133*, 23–46.

Ben-Tal, A., Boyd, S., & Nemirovski, A. (2006). Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Mathematical Programming, 107*(1–2), 63–89.

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

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

Ben-Tal, A., Hertog, D. D., De Waegenaere, A., Melenberg, B., & Rennen, G. (2013). Robust solutions of optimization problems affected by uncertain probabilities. *Management Science, 59*(2), 341–357.

Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming A, 88*, 411–424.

Bertsimas, D., Gupta, V., & Kallus, N. (2013). *Data-driven robust optimization*. ArXiv preprint arXiv: 1401.0212

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

Blanquero, R., Carrizosa, E., & Hansen, P. (2009). Locating objects in the plane using global optimization techniques. *Mathematics of Operations Research, 34*, 837–858.

Bouman, P. C., van den Akker, J. M., & Hoogeveen, J. A. (2011). Recoverable robustness by column generation. In C. Demetrescu, & M. M. Halldórsson (Eds.), *Algorithms – ESA 2011*. In *Lecture Notes in Computer Science: Vol. 6942* (pp. 215–226). Springer Berlin Heidelberg.

Brimberg, J., & Wesolowsky, G. O. (2000). Note: facility location with closest rectangular distances. *Naval Research Logistics (NRL), 47*(1), 77–84.

Brimberg, J., & Wesolowsky, G. O. (2002a). Locating facilities by minimax relative to closest points of demand areas. *Computers & Operations Research, 29*(6), 625–636.

Brimberg, J., & Wesolowsky, G. O. (2002b). Minisum location with closest euclidean distances. *Annals of Operations Research, 111*(1–4), 151–165.

Büsing, C., Koster, A. M. C. A., & Kutschka, M. (2011). Recoverable robust knapsacks: the discrete scenario case. *Optimization Letters, 5*(3), 379–392.

Cacchiani, V., Caprara, A., Galli, L., Kroon, L., Maroti, G., & Toth, P. (2012). Railway rolling stock planning: Robustness against large disruptions. *Transportation Science, 46*(2), 217–232.

Caprara, A., Galli, L., Stiller, S., & Toth, P. (2014). Delay-robust event scheduling. *Operations Research, 62*(2), 274–283.

Carlsson, J. G., Behroozi, M., & Mihic, K. (2015). Wasserstein distance and the distributionally robust TSP. Submitted manuscript.

Carrizosa, E., & Fliege, J. (2002). Generalized goal programming: polynomial methods and applications. *Mathematical Programming, 93*(2), 281–303.

Cicerone, S., D'Angelo, G., Stefano, G., Frigioni, D., & Navarra, A. (2009a). Recoverable robust timetabling for single delay: Complexity and polynomial algorithms for special cases. *Journal of Combinatorial Optimization, 18*, 229–257.

Cicerone, S., D'Angelo, G., Stefano, G. D., Frigioni, D., & Navarra, A. (2007). Robust algorithms and price of robustness in shunting problems. In *Proceedings of the 7th workshop on algorithmic approaches for transportation modeling, optimization, and systems (ATMOS07)* (pp. 175–190).

Cicerone, S., D'Angelo, G., Stefano, G. D., Frigioni, D., Navarra, A., Schachtebeck, M., & Schöbel, A. (2009b). Recoverable robustness in shunting and timetabling. In *Robust and online large-scale optimization*. In *number 5868 in Lecture Notes in Computer Science* (pp. 28–60). Springer.

Dattorro, J. (2010). *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA.

Drezner, Z., Klamroth, K., Schöbel, A., & Wesolowsky, G. O. (2002). The Weber problem. In Z. Drezner, & H. W. Hamacher (Eds.), *Facility location: Applications and theory* (pp. 1–36). Springer.

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

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

Fischetti, M., & Monaci, M. (2009). Light robustness. In R. K. Ahuja, R. Möhring, & C. D. Zaroliagis (Eds.), *Robust and online large-scale optimization*. In *Lecture Note on Computer Science: Vol. 5868* (pp. 61–84). Springer.

Goerigk, M. (2012). *Algorithms and concepts for robust optimization*. University of Göttingen Phd thesis.

Goerigk, M., Deghdak, K., & T'Kindt, V. (2015). A two-stage robustness approach to evacuation planning with buses. *Transportation Research Part B: Methodological, 78*, 66–82.

Goerigk, M., & Schöbel, A. (2010). An empirical analysis of robustness concepts for timetabling. In T. Erlebach, & M. Lübbecke (Eds.), *Proceedings of ATMOS10*. In *OpenAccess Series in Informatics (OASIcs): Vol. 14* (pp. 100–113). Dagstuhl, Germany.

Goerigk, M., & Schöbel, A. (2011). A scenario-based approach for robust linear optimization. In *Proceedings of the first international ICST conference on theory and practice of algorithms in (computer) systems, TAPAS'11* (pp. 139–150). Berlin, Heidelberg: Springer-Verlag.

Goerigk, M., & Schöbel, A. (2014). Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. *Computers & Operations Research, 52*, 1–15. Part A.

Goerigk, M., & Schöbel, A. (2016). Algorithm engineering in robust optimization. In L. Kliemann, & P. Sanders (Eds.), *Algorithm engineering: Selected results and surveys*. In *LNCS State of the Art: Vol. 9220*. Springer. Final Volume for DFG Priority Program 1307.

Guastaroba, G., Mansini, R., & Speranza, M. G. (2009). On the effectiveness of scenario generation techniques in single-period portfolio optimization. *European Journal of Operational Research, 192*(2), 500–511.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research, 165*, 289–306.

Hiriart-Urruty, J. B., & Lemaréchal, C. (1993). *Convex analysis and minimization algorithms*. Berlin: Springer Verlag.

Horst, R., & Thoai, N. V. (1999). Dc programming: Overview. *Journal of Optimization Theory and Applications, 103*(1), 1–43.

Kasperski, A., & Zielinski, P. (2015). Robust recoverable and two-stage selection problems. CoRR, abs/1505.06893.

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

Liebchen, C., Lübbecke, M., Möhring, R. H., Stiller, S., & Zaroliagis, C. D. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In R. K. Ahuja, & R. Möhring (Eds.), *Robust and online large-scale optimization*. In *Lecture Note on Computer Science: Vol. 5868* (pp. 1–27). Springer.

Megiddo, N. (1984). The weighted euclidean 1-center problem. *Mathematics of Operations Research, 8*(4), 498–504.

Nickel, S., & Puerto, J. (2009). *Location theory - a unified approach*. Springer.

Nickel, S., Puerto, J., & Rodriguez-Chia, A. M. (2003). An approach to location models involving sets as existing facilities. *Mathematics of Operations Research, 28*(4), 693–715.

Plastria, F., & Carrizosa, E. (2001). Gauge distances and median hyperplanes. *Journal of Optimization Theory and Applications, 110*(1), 173–182.

Prigent, J.-L. (2007). *Portfolio optimization and performance analysis*. CRC Press.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton: Princeton Landmarks.

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

Sergeyev, Y. D., & Kvasov, D. E. (2010). *Lipschitz Global Optimization*. John Wiley & Sons, Inc.

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

Stiller, S. (2008). *Extending Concepts of Reliability. Network Creation Games, Real-time Scheduling, and Robust Optimization*. PhD thesis, TU Berlin.

Sun, W., & Yuan, Y. (2006). *Optimization theory and methods: nonlinear programming*. Springer.

Toregas, C., Swain, R., ReVelle, C., & Bergman, L. (1971). The location of emergency facilities. *Operations Research, 19*, 1363–1373.

Ward, J. E., Wendell, R. E., & Richard, E. (1985). Using block norms for location modeling. *Operations Research, 33*, 1074–1090.

Witzgall, C. (1964). Optimal location of a central facility: Mathematical models and concepts. *Technical report 8388*. National Bureau of Standards.