View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Digitale Bibliothek Thüringen

Technische Universität Ilmenau
Institut für Mathematik

Preprint No. M 18/11

An algorithmic approach to multiobjective optimization with decision uncertainty

Gabriele Eichfelder, Julia Niebling, Stefan Rocktäschel

07. November 2018

URN: urn:nbn:de:gbv:ilm1-2018200159

Impressum:
Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69-3621
Fax: +49 3677 69-3270
http://www.tu-ilmenau.de/math/

ilmedia

An Algorithmic Approach to Multiobjective Optimization with Decision Uncertainty

Gabriele Eichfelder*, Julia Niebling‖, Stefan Rocktäschel**

November 7, 2018

Abstract

In real life applications optimization problems with more than one objective function are often of interest. Next to handling multiple objective functions, another challenge is to deal with uncertainties concerning the realization of the decision variables. One approach to handle these uncertainties is to consider the objectives as set-valued functions. Hence, the image of one variable is a whole set, which includes all possible outcomes of this variable. We choose a robust approach and thus these sets have to be compared using the so called upper-type less order relation.

We propose a numerical method to calculate a covering of the set of optimal solutions of such an uncertain multiobjective optimization problem. We use a branch-and-bound approach and lower and upper bound sets for being able to compare the arising sets. The calculation of these lower and upper bound sets uses techniques known from global optimization as convex underestimators as well as techniques used in convex multiobjective optimization as outer approximation techniques. We also give first numerical results for this algorithm.

Key Words: Multiobjective Optimization, Decision Uncertainty, Branch-and-Bound Algorithm

Mathematics subject classifications (MSC 2000): 90C29, 90C26, 58C06, 90C31

1 Introduction

In multiobjective optimization one considers optimization problems with more than one objective function. This is already a challenge. Dealing with multiobjective optimization for real life problems can lead to an additional difficulty. Often, the calculated solutions

*Institute for Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, gabriele.eichfelder@tu-ilmenau.de

‖Institute for Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, julia.niebling@tu-ilmenau.de

**Institute for Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, stefan.rocktaeschel@tu-ilmenau.de

cannot be used precisely, because they can only be realized within a certain accuracy. This is for instance the case, when a magnet system for a measurement technique for electrically conducting fluids should be constructed, see [10]. There, the optimal direction of the magnetization of each magnet and the optimal magnetization have to be determined such that the so called Lorentz force is maximized and the weight of the system is minimized. In practice an (optimally) chosen magnetic direction cannot be realized in any arbitrary accuracy, as magnets can only be produced within some tolerance. Therefore, decision uncertainty has to be taken into account. Another example is the growing media mixing problem for a plant nursery, see [10, 21, 20], where a mixture of peat and compost for the growing media has to be determined, which can also not be mixed exactly by workers.

Also in case of such uncertainties in the realization of variables, the actual realized solutions should lead to near-optimal values. This kind of uncertainty in optimization is called *decision uncertainty*, which should be distinguished from *parameter uncertainty*, because the inaccuracies are caused by the decision variables, [10]. Parameter uncertainty was considered in several works, for example in [3] in the single-objective case or in [11, 12, 23] for the multiobjective case.

Decision uncertainty for single-objective optimization problems has been handled with minmax robustness under different names likes 'robust optimization with implementation error', e.g. in [4], or robust regularization, e.g. in [24]. In multiobjective optimization there are also different approaches to treat uncertainties, for example sensitivity analysis, [2], or evaluating the mean or integral of each objective over the set of possible values of a solution, [7], or adding a robustness measure as a new objective function, [32]. We want to follow the so-called worst-case robustness approach as it was done in [10], see also [19].

In the worst case approach one considers all possible outcomes, which leads to sets which have to be compared. In the single-objective case these sets are just intervals, which can be compared much easier numerically. In case of a multiobjective optimization problem with m objectives we have to compare subsets in $\mathbb{R}^m$. This means we have to solve a specific set-valued optimization problem with a certain set-order relation to handle the uncertainties.

In set-valued optimization different possibilities to compare sets are discussed in the literature. In case of the worst-case approach we have to use the upper-type less order relation, see for instance [22, 16]. When comparing whole sets one also speaks of the set approach in set-valued optimization. So far there exists only a limited number of numerical algorithms to solve such set optimization problems. For unconstrained set-valued optimization and a similar order relation Löhne and Schrage introduced an algorithm in [26], which is applicable for linear problems only. Jahn presented some derivative-free algorithms, see [15] to find one single solution of the whole set of optimal solutions in case the sets which have to be compared are convex. Köbis and Köbis extended the method from [15] to the nonconvex case, i.e. when the sets are nonconvex sets, see [18]. However, all methods aim at finding one single minimal solution only. In [13] for the first time a method for nonconvex sets is presented, which uses discretization to compare sets and which can find many minimal solutions, but still not all. The procedure was parallelized and implemented on a CPU and GPU. As set-valued optimization problems have in general infinitely many optimal solutions, a representation of the whole set of optimal solutions is of interest.

In [10], see also [21, 20], for some specific multiobjective decision uncertain optimization problems solution approaches or characterizations of the optimal solution set have been provided. But they are all for specific cases only, as for linear or for monotone objective functions.

In this paper we suggest a numerical approach for smooth multiobjective optimization problems with decision uncertainty with the worst case approach under quite general assumptions. Solving such problems is the same as solving a specific set-valued optimization problem where the image sets are nonconvex. Thus, our approach also gives new ideas for developing algorithms for solving set optimization problems with the set approach, even in case the sets which have to be compared are nonconvex. The proposed algorithm determines a covering of the set of optimal solutions, i.e. a subset of the feasible set which contains all optimal solutions.

Thereby, we use the branch-and-bound concept to partition the feasible set and to detect regions of the feasible set which do not contain optimal solutions quickly. We have to assume that the feasible set is a subset of a box which is defined by convex constraints, and that the set of uncertainties is convex as well, to be able to solve the arising subproblems numerically. However, we do not need any assumptions on the convexity of the objective functions. As we are using techniques from global optimization, these functions are allowed to be nonconvex.

The paper is structured as follows: In Section 2 the preliminaries for multiobjective optimization and decision uncertainty can be found. Additionally, the relations to set optimization are shown. The main ideas for our algorithm are explained in Section 3. In particular, we mention the concept of convex underestimators and concave overestimators. Both concepts are essential to find lower and upper bound sets. The description how those bound sets can be determined can be found in Section 3.2 and 3.3. The whole branch-and-bound algorithm as well as some numerical results are presented in Section 4. We end with a conclusion and an outlook in Section 5.

2 Multiobjective optimization with decision uncertainty

We are starting by introducing the multiobjective optimization problem for a nonempty feasible set $\Omega \subseteq \mathbb{R}^n$ and twice continuously differentiable functions $f_j\colon \Omega \to \mathbb{R}$, $j = 1, \ldots, m$:

$$\min_{x \in \Omega} f(x). \tag{P}$$

Without any uncertainties one says that a point $x^* \in \Omega$ is efficient for (P), if there is no $x \in \Omega$ with $f(x) \neq f(x^*)$ and $f(x) \leq f(x^*)$. Here, we use for two vectors $a, b \in \mathbb{R}^m$ the notation $a \leq b \;:\Leftrightarrow a_j \leq b_j$ for all $j = 1, \ldots, m$. If x^* is efficient for (P), then we call $y^* := f(x^*)$ to be a nondominated point of (P).

Next, we introduce the relevant notions from multiobjective optimization with decision uncertainty before we shortly point out the relations to the more general class of set optimization problems.

2.1 Decision uncertainty

As described in the introduction it is possible that the realization of the decision variable x is associated with uncertainties. These uncertainties will be modeled by a convex and compact set $Z \subseteq \mathbb{R}^n$ with $0 \in Z$. Hence, instead of x it might happen that $x + z$ for some $z \in Z$ is realized due to the uncertainties. We follow the notation as introduced in [10].

In view of robustness, only feasible points $x \in \Omega$ which are feasible for any uncertainty $z \in Z$ are of interest. This reduces the feasible set to the set of *decision robust feasible points* defined by

$$S := \{x \in \Omega \mid x + z \in \Omega \text{ for all } z \in Z\}.$$

It is a well known challenge in robust optimization to calculate S from Ω and Z, also in the single-objective case. In this work, we assume that S is known and nonempty and we only concentrate on those challenges which are due to the multiple objectives.

Moreover, we assume that S is convex and that there exists a box X which contains S. A set $X \subseteq \mathbb{R}^n$ is called a *box* (or hyper rectangle) if $X = \{x \in \mathbb{R}^n \mid \underline{x}_i \leq x \leq \overline{x},\ i = 1, \dots, n\}$ with two points $\underline{x}, \overline{x} \in \mathbb{R}^n$ with $\underline{x} \leq \overline{x}$. The set of all n-dimensional real boxes will be denoted by $\mathbb{IR}^n$. These requirements are due to the fact that we will have to solve single-objective subproblems with a convex objective function over the set S. For being able to solve those problems globally and efficiently we need the assumption of convexity of S. As our algorithm works with a subdivision in the pre-image space we also need the structure of the box.

For defining decision robust efficient solutions we have to take for each x all possible realizations of x into account, i.e., the set $\{x + z \mid z \in Z\}$. As we are interested in the values of the objective functions we thus have to compare the sets

$$f_Z(x) := \{f(x + z) \in \mathbb{R}^m \mid z \in Z\} \text{ for all } x \in S$$

which defines a set-valued map $f_Z \colon S \rightrightarrows \mathbb{R}^m$.

In case the functions f_i are linear, then we have $f_Z(x) = \{f(x)\} + \{f(z) \in \mathbb{R}^m \mid z \in Z\}$. This simplifies the problem significantly, see [10, Theorem 23] or [9]. In case the functions f_i and the set Z are convex, the sets $f_Z(x)$ for $x \in S$ do not have be convex, as the following example shows:

Example 2.1 *Let $f_1, f_2 \colon \mathbb{R}^2 \to \mathbb{R}$ with $f_1(x) = x_1, f_2(x) = x_1^2 + x_2^2$ and $Z = [-0.1, 0.1] \times [0, 0.1]$ be given. Then for $x = 0$ the set*

$$f_Z(0) = \{f(0 + z) \mid z \in Z\} = \left\{ \begin{pmatrix} z_1 \\ z_1^2 + z_2^2 \end{pmatrix} \in \mathbb{R}^2 \mid z \in Z \right\}$$

contains the points $y^1 = (-0.1, 0.02)$ and $y^2 = (0.1, 0.02)$ but not the point $0.5y^1 + 0.5y^2 = (0, 0.02)$. This is because $f(0 + z) = (0, 0.02)$ holds only for $z = (0, -\sqrt{0.02})$ or $z = (0, \sqrt{0.02})$. But in both cases it is $z \notin Z$. Hence, the set $f_Z(0)$ is not convex.

Motivated by the definition of optimality for single-objective decision uncertain optimization as well as by the definition of optimality in parameter uncertain multiobjective optimization, the following optimality concept for the problem (P) w.r.t. uncertainty given by the set Z was introduced in [10]:

Definition 2.2 *A point $x^* \in S$ is called a decision robust strictly efficient solution of (P) w.r.t. Z if there is no $x \in S \setminus \{x^*\}$ with the property*

$$f_Z(x) \subseteq f_Z(x^*) - \mathbb{R}^m_+.$$

We illustrate this definition with the following example:

Example 2.3 *Assume $S = \{x^1, x^2\}$ and Z as well as $f\colon S \to \mathbb{R}^2$ are in such a way that the sets $f_Z(x^1)$ and $f_Z(x^2)$ look as in Figures 1 and 2, respectively. Then for the situation in Figure 1, only x^1 is a decision robust strictly efficient solution w.r.t. Z, while for Figure 2 both points, i.e. x^1 and x^2, are decision robust strictly efficient solutions.*

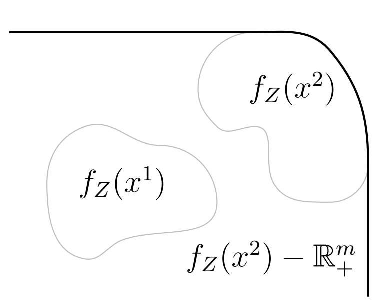


Figure 1: Example 2.3, first case

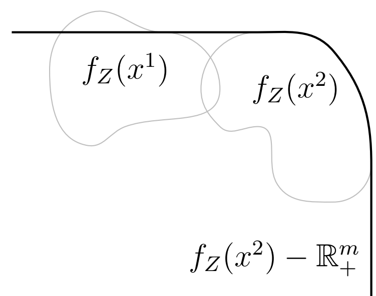


Figure 2: Example 2.3, second case

As one can see from the definition, one has to be able to verify whether it holds $A \subseteq B - \mathbb{R}^m_+$ for two sets $A, B \subseteq \mathbb{R}^m$ numerically. In case of polyhedral sets this can be done by using a finite number of support functionals, see [15]. In case of arbitrary closed convex sets one might need an infinite number of such linear functionals. Additionally, one has to solve a minimizing problem for each of these functionals and for each of the sets to decide whether the subset condition holds. This is based on the equivalence

$$A \subseteq B - \mathbb{R}^m_+ \qquad \Leftrightarrow \qquad \forall\, \ell \in \mathbb{R}^m_+\colon\ \sup_{a \in A} \ell^T a \leq \sup_{b \in B} \ell^T b$$

which was formulated in a more general setting in [14, Theorem 2.1].

2.2 Relation to set optimization

In set optimization one studies set-valued optimization problems. Closely related to determining decision robust strictly efficient solutions is solving the following set optimization problem:

$$\min_{x \in S} f_Z(x) = \{f(x+z) \mid z \in Z\}. \qquad (SOP)$$

There are different approaches to define optimal solutions for problems like (SOP). For an introduction to set optimization see the book [17]. In case of the set approach, see [16], one uses order relations to compare the sets which are the images of the objective function. As we are minimizing in our multiobjective optimization problem, the 'maximal elements' of a set are in some sense the worst elements which fits to our worst-case approach. Comparing the 'maximal elements' of a set corresponds to the upper-less order relation, which we introduce next.

Definition 2.4 *[16] Let $A, B \subseteq \mathbb{R}^m$ be two nonempty sets. The upper-type (u-type) less order relation $\preccurlyeq_u$ is defined by:*

$$A \preccurlyeq_u B \ \Leftrightarrow \ A \subseteq B - \mathbb{R}^m_+.$$

For nonempty sets $A, B \subseteq \mathbb{R}^m$ we have

$$A \preccurlyeq_u B \Leftrightarrow \forall a \in A \ \exists b \in B : a \leq b.$$

Moreover, the order relation is reflexive and transitive, but it is in general not anti-symmetric. However, it holds

$$A \preccurlyeq_u B \text{ and } B \preccurlyeq_u A \Leftrightarrow A - \mathbb{R}^m_+ = B - \mathbb{R}^m_+.$$

Now we can state the definition of an optimal solution of a set-valued optimization problem as in (SOP). For this we use a definition which was formulated in [29]:

Definition 2.5 *Let a nonempty set $X \subseteq \mathbb{R}^n$ and a set-valued map $H : X \rightrightarrows \mathbb{R}^m$ be given with $H(x) \neq \emptyset$ for all $x \in X$. A point $x^* \in X$ is called a strictly optimal solution of the set optimization problem*

$$\min_{x \in X} H(x)$$

w.r.t. $\preccurlyeq_u$ if there exists no $x \in X \setminus \{x^*\}$ with $H(x) \preccurlyeq_u H(x^*)$.

In our case we have the special set-valued map described by $f_Z \colon S \rightrightarrows \mathbb{R}^m$ with $f_Z(x) = \{f(x + z) \mid z \in Z\}$. Obviously, a point $x^* \in S$ is a decision robust strictly efficient solution of (P) w.r.t. Z if and only if $x^* \in S$ is a strictly optimal solution of the set optimization problem (SOP). Hence, we present in this paper an algorithm to calculate a covering of the set of strictly optimal solutions of a specific set optimization problem. Thus, these techniques can also be used to develop algorithms for more general set optimization problems.

A basic technique of our algorithm will be a branch-and-bound approach. For the bounding step lower and upper bounds will be important. The next definition clarifies these terms.

Definition 2.6 *Let $A \subseteq \mathbb{R}^m$ be a nonempty set.*

- *A set $U \subseteq \mathbb{R}^m$ is called an upper bound set/upper bound for A if $A \preccurlyeq_u U$.*
- *A set $L \subseteq \mathbb{R}^m$ is called a lower bound set/lower bound for A if $L \preccurlyeq_u A$.*

3 Algorithmic approach

Our algorithm uses the concept of a branch-and-bound method. The branching will be in the pre-image space $\mathbb{R}^n$. We have assumed that there is a box X which contains the convex feasible set S. This is for instance the case when S is given by convex inequality constraints and by lower and upper bounds for each variable. We start with the box X and partition it along the longest edge into two subboxes, see for instance [28] for a more detailed description. On each subbox X^* we test whether a sufficient criteria is satisfied that $X^* \cap S$ does not contain a decision robust strictly efficient solution of (P) w.r.t. Z. In case such a criteria is satisfied then we do not consider this subbox and the feasible points $X^* \cap S$ anymore. Otherwise, we partition the box until all boxes are either discarded or smaller than a predefined value.

For such a branch-and-bound scheme a good criteria for discarding a box is essential. These criteria are in general based on lower bounds obtained on the subboxes and on upper bounds obtained within the procedure. This is a widely used approach in single-objective global (i.e. nonconvex) optimization. There, the upper bounds are function values of feasible solutions and the lower bounds are bounds for all possible values of the objective over a subbox which are determined by interval arithmetic or by other underestimation methods. Hence, just scalars have to be compared. In our setting, already a 'function value' $f_Z(x)$ for some feasible x is a whole set. As these lower and upper bounds have to be compared frequently within such an algorithm, we will present a way to avoid to compare sets as $f_Z(x)$ for some x directly and will present replacements (i.e. sufficient conditions) with points and sets which have a very simple structure. To distinguish between upper and lower bound sets in our article the corresponding variables x, z and subboxes or subsets of X and Z are indicated with $\tilde{\cdot}$ or $\cdot^*$, respectively. This means that an upper bound set is computed with respect to a fixed point $\tilde{x} \in \tilde{X}$ and a lower bound set is determined for all sets $f_Z(x^*) = \{f(x+z) \mid x \in X^*, z \in Z\}$ with $x^* \in X^*$.

As the objective functions f_i and also the sets $f_Z(x)$ are not necessarily convex, we will use the concept of convex relaxations for being able to formulate such replacements and a numerically tractable sufficient condition finally.

3.1 Convex underestimators and concave overestimators

As shown with Example 2.1, the sets $f_Z(x)$ for $x \in S$ may be nonconvex, even in case the functions $f_i\colon S \to \mathbb{R}$ are convex. For that reason we will make use of the concept of *convex underestimators* and *concave overestimators*, respectively, depending on whether we aim at lower or at upper bounds. Let $X \subseteq \mathbb{R}^n$ be a box and $h\colon X \to \mathbb{R}$ a function. Then a convex underestimator of h on X is a function $\underline{h}\colon X \to \mathbb{R}$ which is convex and with $\underline{h}(x) \leq h(x)$ for all $x \in X$. A simple way to construct such convex underestimators is explained next and is known in the literature as αBB method.

Let the box be defined by $X = [\underline{x}, \overline{x}] \in \mathbb{IR}^n$. Then we obtain a convex underestimator of a smooth function h by

$$\underline{h}(x) := h(x) + \frac{\alpha_h}{2}(\underline{x} - x)^T(\overline{x} - x),$$

where $\alpha_h \geq \max\{0, -\min_{x \in X} \lambda_{\min,h}(x)\}$. Here, $\lambda_{\min,h}(x)$ denotes the smallest eigenvalue of the Hessian $H_h(x)$ of h in x, [27]. A lower bound for $\lambda_{\min,h}(x)$ over X can easily be calculated with the help of interval arithmetic, see [27]. In our algorithm we use the Matlab toolbox Intlab [30] for these calculations. See also [31] for improved lower bounds. There are also other possibilities for the calculation of convex underestimators. For example in [1] special convex underestimators for bilinear, trilinear, fractional, fractional trilinear or univariate concave functions were defined. Here, we restrict ourselves to the above proposed convex underestimator. The theoretical results remain true in case the above underestimators are replaced by tighter ones. Another important benefit is stated in the following remark, where $\omega(X)$ denotes the box width of X, i.e. $\omega(X) := \|\overline{x} - \underline{x}\|_2$.

Remark 3.1 *[1] For all $\alpha_h \geq 0$ the maximal pointwise difference between h and $\underline{h}$ is $\frac{\alpha_h}{2}\omega(X)^2$, i.e. $\max\limits_{x\in X} |h(x) - \underline{h}(x)| = \frac{\alpha_h}{2}\omega(X)^2$.*

In nonconvex optimization one uses the minimum value of $\underline{h}$ over $X \cap S$ as a lower bound for the values of h on $X \cap S$. The minimum value can be calculated by standard techniques from convex optimization in case the set S is convex. We use convex underestimators to be able to numerically calculate the elements of sets $L \subseteq \mathbb{R}^m$ such that

$$L \subseteq f_Z(x) - \mathbb{R}^m_+ \;\Leftrightarrow\; L \preccurlyeq_u f_Z(x)$$

holds for all $x \in X^* \cap S$ for some subbox X^* of X.

Also concave overestimators of a function h on X are of interest, i.e. concave functions $\overline{h}\colon X \to \mathbb{R}$ with $\overline{h}(x) \geq h(x)$ for all $x \in X$. If we calculate a convex underestimator of the function $-h$ as described above, i.e.

$$\underline{-h}(x) := -h(x) + \frac{\alpha_{-h}}{2}(\underline{x} - x)^T(\overline{x} - x),$$

where $\alpha_{-h} \geq \max\{0, -\min_{x\in X} \lambda_{\min,-h}(x)\}$, then $\overline{h} := -(\underline{-h})$ is such a concave overestimator of h on X. We use such concave overestimators to be able to numerically calculate sets $U \subseteq \mathbb{R}^m$ such that

$$f_Z(\tilde{x}) \subseteq U - \mathbb{R}^m_+ \;\Leftrightarrow\; f_Z(\tilde{x}) \preccurlyeq_u U$$

holds for some given $\tilde{x} \in S$. The advantage is that while it might be numerically difficult to compare the lower bound L of all sets $f_Z(x)$ for all $x \in X^* \cap S$ with $f_Z(\tilde{x})$, it might be much easier to compare L with U. Note that we have

$$U \subseteq L - \mathbb{R}^m_+ \;\Rightarrow\; f_Z(\tilde{x}) \subseteq f_Z(x) - \mathbb{R}^m_+ \text{ for all } x \in X^* \cap S.$$

When we use the set order relation as defined in Definition 2.4 we can write this equivalently as

$$\begin{aligned} U \preccurlyeq_u L \quad &\Rightarrow \quad f_Z(\tilde{x}) \preccurlyeq_u U \preccurlyeq_u L \preccurlyeq_u f_Z(x) \text{ for all } x \in X^* \cap S \\ &\Rightarrow \quad f_Z(\tilde{x}) \preccurlyeq_u f_Z(x) \text{ for all } x \in X^* \cap S. \end{aligned} \tag{1}$$

The implication holds as $\preccurlyeq_u$ is a transitive set order relation. Thus, in case $\tilde{x} \in S \setminus X^*$, we can discard the subbox X^* as it does not contain any decision robust strictly efficient solution of (P) w.r.t. Z.

3.2 Upper bound sets

First, we want to calculate a set U with $f_Z(\tilde{x}) \preccurlyeq_u U$, i.e. to a given box $\tilde{X}$ we fix one point $\tilde{x} \in \tilde{X} \cap S$. Hence U has to be only an upper bound for one image of the set-valued map f_Z. To begin with, we explain how we can construct such a set U which is a singleton.

Then we describe how this upper bound can be improved by using outer approximations as known from convex multiobjective optimization.

A simple upper bound for a set is the so called *anti-ideal* point. For the set $f_Z(\tilde{x}) = \{f(\tilde{x}+z) \mid z \in Z\}$ this is the point $\overline{a}$ defined by

$$\overline{a}_j := \max_{y \in f_Z(\tilde{x})} y_j = \max_{z \in Z} f_j(\tilde{x}+z) \text{ for all } j = 1, \ldots, m.$$

Hence, the anti-ideal point can easily be determined if f_j is a concave function for $j = 1, \ldots, m$, as Z is assumed to be a convex and compact set and the functions f_j are twice continuously differentiable. One can apply any solution method from single-objective constrained optimization as for instance an SQP method.

In case f_j is not concave such a local solution method as SQP might only deliver a locally maximal solution and not a globally maximal one. In that case we use the concave overestimators which were introduced in Subsection 3.1. The result is summarized in the next lemma. This lemma needs a box $\hat{Z}$ with $Z \subseteq \hat{Z}$, but recall that Z was assumed to be a compact convex set and thus such a set $\hat{Z}$ can easily be calculated. The reason for the assumption of a box is that we can only determine concave overestimators over boxes as explained in Subsection 3.1. With Remark 3.1 it follows that a small box $\hat{Z}$ leads to a tighter concave overestimator. Therefore, $\hat{Z}$ should be chosen as small as possible.

Lemma 3.2 *Let $\hat{Z} \in \mathbb{IR}^n$ be a box with $Z \subseteq \hat{Z}$ and let $\tilde{x} \in S$ be given. Let $\overline{f}_j$ be the concave overestimator of f_j on the box $\{\tilde{x}\} + \hat{Z}$ for all $j = 1, \ldots, m$. The singleton set U with*

$$U := \{\bar{p}\} \text{ with } \bar{p} := (\max_{z \in Z} \overline{f}_1(\tilde{x}+z), \ldots, \max_{z \in Z} \overline{f}_m(\tilde{x}+z))^T \tag{2}$$

is an upper bound set for $f_Z(\tilde{x})$, i.e. $f_Z(\tilde{x}) \preccurlyeq_u U$.

Proof. To show $f_Z(\tilde{x}) \preccurlyeq_u U$ it has to hold $f_Z(\tilde{x}) \subseteq U - \mathbb{R}^m_+$. Let $w \in f_Z(\tilde{x})$ be arbitrary chosen, i.e. there is a $z \in Z$ with $w = f(\tilde{x}+z)$. As $\overline{f}_j$ is a concave overestimator of f_j on $\{\tilde{x}\} + \hat{Z}$ and $z \in Z \subseteq \tilde{Z}$ we obtain for every $j = 1, \ldots, m$:

$$w_j = f_j(\tilde{x}+z) \leq \overline{f}_j(\tilde{x}+z) \leq \max_{z \in Z} \overline{f}_j(\tilde{x}+z).$$

Therefore $w \in U - \mathbb{R}^m_+$. □

The optimization problems in (2) have a convex and compact feasible set and twice continuously differentiable concave objective functions, which are maximized. Thus they can be solved for instance with an SQP method. Lemma 3.2 uses that it holds for the set

$$\overline{f_Z}(\tilde{x}) := \{(\overline{f}_1(\tilde{x}+z), \ldots, \overline{f}_m(\tilde{x}+z))^T \mid z \in Z\}$$

that $f_Z(\tilde{x}) \subseteq \overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$ and that $\overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$ is a convex set. Thus, the anti-ideal point of $\overline{f_Z}(\tilde{x})$ can be calculated by known local methods for single-objective optimization. Figure 3 shows the idea on how to obtain the upper bound set U.

This rough upper bound can be improved by using outer approximation techniques from convex multiobjective optimization. These can be applied as the set $\overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$ is a convex set. The algorithm which we are using is called Benson's outer approximation

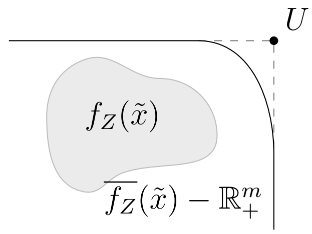


Figure 3: The set $f_Z(\tilde{x})$ and its singleton upper bound set U according to Lemma 3.2.

algorithm and was introduced for linear multiobjective optimization in [5] and extended to the convex case in [8] and then also in [25]. The idea of Benson's outer approximation algorithm is to construct supporting hyperplanes of the set $\overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$. One step depends on one fixed point $\bar{p}$, which can for instance be the anti-ideal point. Within Benson's outer approximation algorithm one solves a single-objective convex optimization problem of the following type:

$$\min_{(z,t)\in\mathbb{R}^n\times\mathbb{R}} t \text{ s.t. } z \in Z,\ \bar{p} - te \le \overline{f}(\tilde{x} + z). \qquad (P_{\tilde{x},\bar{p}})$$

With $e \in \mathbb{R}^m$ we denote the m-dimensional all-one vector $(1, \dots, 1)^T \in \mathbb{R}^m$.

Let $(\tilde{z}, \tilde{t})$ be an optimal solution of $(P_{\tilde{x},\bar{p}})$ and let $\tilde{\lambda} \ge 0$ be a Lagrange multiplier to the constraint $\bar{p} - te \le \overline{f}(\tilde{x} + z)$. Then

$$\{y \in \mathbb{R}^m \mid \tilde{\lambda}^T y = \tilde{\lambda}^T(\bar{p} - \tilde{t}e)\}$$

describes a supporting hyperplane of $\overline{f_Z}(\tilde{x})$. In case no Lagrange multiplier is available, a single-objective linear optimization problem can be solved to calculate a normal vector $\tilde{\lambda}$ of the supporting hyperplane, see [8].

Note that already the anti-ideal point of $\overline{f_Z}(\tilde{x})$ gives m supporting hyperplanes of $\overline{f_Z}(\tilde{x})$ by

$$\{y \in \mathbb{R}^m \mid y_j = \max_{z\in Z} \overline{f}_j(\tilde{x} + z)\}$$

for every $j = 1, \dots, m$. Several such supporting hyperplanes can be constructed to various points $\bar{p}$ with the help of $(P_{\tilde{x},\bar{p}})$. In our numerical experiments we limited ourselves to the hyperplane which we obtain for $\bar{p}$ as the anti-ideal point of $\overline{f_Z}(\tilde{x})$ and to those which we obtain directly from the anti-ideal point of $\overline{f_Z}(\tilde{x})$.

The construction of such supporting hyperplanes is explained in detail in [8, 25]. This technique is also used for solving nonconvex multiobjective optimization problems in [28]. In that paper one can also find more details on the construction of improved bounds by using Benson's outer approximation technique. Adding more hyperplanes to get a better outer approximation is possible, but then even more single-objective convex optimization problems have to be solved. Moreover, the calculation of the intersection of these hyperplanes gets more challenging. Also steering the calculation of the hyperplanes within the algorithm by an adaptive choice of the points $\bar{p}$ in $(P_{\tilde{x},\bar{p}})$ is an interesting approach for further improvements of the proposed algorithmic approach. Such an approach was for instance followed in [28] for nonconvex multiobjective optimization problems.

The next lemma gives a summary of the construction of our upper bound set, which is also illustrated with Figure 4.

Lemma 3.3 *Let $\hat{Z} \in \mathbb{IR}^n$ be a box with $Z \subseteq \hat{Z}$ and let $\tilde{x} \in S$ be given. Let $\overline{f}_j$ be the concave overestimator of f_j on the box $\{\tilde{x}\} + \hat{Z}$ for all $j = 1, \ldots, m$, and let $\bar{p} \in \mathbb{R}^m$ be the anti-ideal point of the concave overestimator $\overline{f}$ on $\{\tilde{x}\} + Z$, see Lemma 3.2. Moreover, denote the minimal solution of $(P_{\tilde{x},\bar{p}})$ by $(\tilde{z}, \tilde{t})$ and a Lagrange multiplier to the constraint $\bar{p} - te \leq \overline{f}(\tilde{x} + z)$ by $\tilde{\lambda}$. Then the set U with*

$$U := \{y \in \mathbb{R}^m \mid y_j \leq \bar{p}_j,\ j = 1, \ldots, m,\ \tilde{\lambda}^T y = \tilde{\lambda}^T(\bar{p} - \tilde{t}e)\} \tag{3}$$

is an upper bound set for $f_Z(\tilde{x})$, i.e. $f_Z(\tilde{x}) \preccurlyeq_u U$.

Proof. We denote the hyperplane to which $\tilde{\lambda} \geq 0$ is the normal vector by

$$U^* := \{y \in \mathbb{R}^m \mid \tilde{\lambda}^T y = \tilde{\lambda}^T(\bar{p} - \tilde{t}e)\}.$$

As $(\tilde{z}, \tilde{t})$ is feasible for $(P_{\tilde{x},\bar{p}})$ and by the definition of $\bar{p}$ we get for each $j = 1, \ldots, m$:

$$\tilde{t} \geq \bar{p}_j - \overline{f}_j(\tilde{x} + \tilde{z}) \geq 0.$$

As a consequence, we have $\bar{p} - \tilde{t}e \in U$. In particular, $\bar{p} - \tilde{t}e \in U^*$ holds and thus $\bar{p} \in U^* + \mathbb{R}^m_+$.

Next, we want to show that $f_Z(\tilde{x}) \subseteq U^* - \mathbb{R}^m_+$. This follows from [25] as $(P_{\tilde{x},\bar{p}})$ is the same optimization problem which is solved to obtain a supporting hyperplane of $-\overline{f_Z}(\tilde{x}) + \mathbb{R}^m_+$ with the ideal point $-\bar{p}$ (see Subsection 3.3 for more details on the ideal point). There, U^* is by construction a supporting hyperplane of $-\overline{f_Z}(\tilde{x}) + \mathbb{R}^m_+$. To receive a supporting hyperplane of $\overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$ the hyperplane has to be moved to $\bar{p} - \tilde{t}e$. Hence, U^* is a supporting hyperplane of $\overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+$. As $\overline{f}_j$ is a concave overestimator of f_j on the box $\{\tilde{x}\} + \hat{Z}$ for all $j = 1, \ldots, m$, it follows:

$$f_Z(\tilde{x}) \subseteq \overline{f_Z}(\tilde{x}) - \mathbb{R}^m_+ \subseteq U^* - \mathbb{R}^m_+.$$

Now, let $w \in f_Z(\tilde{x})$ be arbitrarily chosen. Then

$$w \in U^* - \mathbb{R}^m_+ = \{y \in \mathbb{R}^m \mid \tilde{\lambda}^T y \leq \tilde{\lambda}^T(\bar{p} - \tilde{t}e)\}$$

and $\bar{p} \in U^* + \mathbb{R}^m_+ = \{y \in \mathbb{R}^m \mid \tilde{\lambda}^T y \geq \tilde{\lambda}^T(\bar{p} - \tilde{t}e)\}$. Thus, there exists some $\mu \in [0, 1]$ with $y := \mu\bar{p} + (1 - \mu)w \in U^*$. By Lemma 3.2 we have $w \leq \bar{p}$. Therefore,

$$w \leq w + \mu(\bar{p} - w) = y = \mu\bar{p} + (1 - \mu)w \leq \bar{p}$$

and $w \leq y$ with $y \in U^* \cap (\{\bar{p}\} - \mathbb{R}^m_+) = U$. Hence, we derive $f_Z(\tilde{x}) \subseteq U - \mathbb{R}^m_+$. □

3.3 Lower bound sets

For a subbox X^* of X we denote the feasible set of X^* by

$$S^* := X^* \cap S.$$

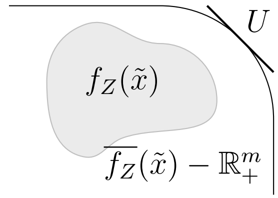


Figure 4: The set $f_Z(\tilde{x})$ and its upper bound set U according to Lemma 3.3.

For applying the implication (1) we propose a method to determine a lower bound L with a simple structure and with

$$L \preccurlyeq_u f_Z(x) \text{ for all } x \in S^* \Leftrightarrow L \subseteq f_Z(x) - \mathbb{R}^m_+ \text{ for all } x \in S^*.$$

Similarly to the anti-ideal point from Lemma 3.2 we can use here the concept of the *ideal point* of a set. While one could use the ideal point of the set

$$\{f(x+z) \in \mathbb{R}^m \mid x \in S^*,\ z \in Z\},$$

already the ideal point of any set

$$f(S^* + z^*) := \{f(x+z^*) \in \mathbb{R}^m \mid x \in S^*\}$$

for any $z^* \in Z$ delivers a lower bound:

Lemma 3.4 *Let $z^* \in Z$, X^* be a subbox of X, and let $a \in \mathbb{R}^m$ be defined by*

$$a_j := \min\{f_j(x+z^*) \in \mathbb{R} \mid x \in S^*\},\ j = 1, \ldots, m. \tag{4}$$

Then the set $L := \{a\}$ is a lower bound set of $f_Z(x)$ for all $x \in S^*$.

Proof. Let $x \in S^*$ be arbitrarily chosen. We have to show that $a \in f_Z(x) - \mathbb{R}^m_+$. As $a_j \leq f_j(x+z^*)$ for all $j = 1, \ldots, m$, we have that $a \in \{f(x+z^*)\} - \mathbb{R}^m_+ \subseteq f_Z(x) - \mathbb{R}^m_+$. □

In case the functions f_j, $j = 1, \ldots, m$ are convex, the single-objective optimization problems in (4) can be solved by any local solution method as an SQP method. Otherwise, convex underestimators, see Subsection 3.1, can be used. Let $\underline{f}$ be the convex underestimator of f on the box X^* (componentwise). Then we can choose one or several points z^* and determine the ideal point of the set

$$\underline{f}(S^* + z^*) := \{\underline{f}(x+z^*) \in \mathbb{R}^m \mid x \in S^*\}$$

for each chosen z^*. As each ideal point gives a lower bound set of the sets $f_Z(x)$ for all $x \in S^*$, also the set of all ideal points to the various points z^* is a lower bound set of $f_Z(x)$ for all $x \in S^*$:

Lemma 3.5 *Let X^* be a subbox of X and let $\underline{f}$ be the componentwise convex underestimator of f on X^*. Let $Z^* = \{z^1, \ldots, z^p\} \subseteq Z$ and for $k = 1, \ldots, p$ determine the ideal point $\underline{a}^k$ of $\underline{f}(S^* + z^k)$, i.e. calculate*

$$\underline{a}^k := (\min_{x \in S^*} \underline{f_1}(x+z^k), \ldots, \min_{x \in S^*} \underline{f_m}(x+z^k))^T$$

Then it holds:

$$L := \{\underline{a}^1, \ldots, \underline{a}^p\} \preccurlyeq_u f_Z(x) \text{ for all } x \in S^*,$$

i.e. L is a lower bound set for all sets $f_Z(x)$ with $x \in S^*$.

Proof. We have to show that for any $k \in \{1, \ldots, p\}$ we have $a^k \in f_Z(x^*) - \mathbb{R}^m_+$ for all $x^* \in S^*$. Thus let $k \in \{1, \ldots, p\}$ and $x^* \in S^*$ be arbitrarily chosen. As

$$\underline{a_j}^k = \min_{x \in S^*} \underline{f_j}(x + z^k) \leq \underline{f_j}(x^* + z^k) \leq f_j(x^* + z^k)$$

for all $j = 1, \ldots, m$, we have that $\underline{a}^k \in \{f(x^* + z^k)\} - \mathbb{R}^m_+ \subseteq f_Z(x^*) - \mathbb{R}^m_+$. □

A visualization of this lemma can be found in Figure 5. The set $f_Z(S^*)$ is defined by $\{f_Z(x + z) \mid x \in S^*, z \in Z\}$. Note that in our algorithm the case of no uncertainty should be considered during the calculation of L. This means we require $0 \in Z^*$.

For the algorithm a large p, i.e. many elements $\{z^1, \ldots, z^p\}$, improves the lower bound set. On the other hand, a large p implies that we have to solve on each subbox a large number of convex single-objective optimization problems to determine the ideal points $\underline{a}^k$, $k = 1, \ldots, p$. We explore this issue in Subsection 4.2.

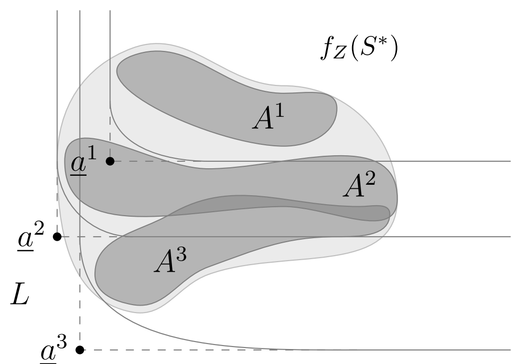


Figure 5: The set $f_Z(S^*)$, the sets $A^k := f(S^* + z^k),\ k = 1, 2, 3$ and its lower bound set $L = \{\underline{a}^1, \underline{a}^2, \underline{a}^3\}$ according to Lemma 3.5.

3.4 Discarding Test

The main idea of our discarding test is the implication given in (1) together with the way in which we construct the sets L and U. We summarize our discarding test in the next theorem:

Theorem 3.6 *Let X^* be a subbox of X, $\tilde{x} \in S$ with $\tilde{x} \notin X^*$, and let the set U be defined as in Lemma 3.3 and L as in Lemma 3.5. If $U \preccurlyeq_u L$, the subbox X^* can be discarded, i.e. $X^* \cap S$ does not contain any decision robust strictly efficient solution of (P) w.r.t. Z.*

Proof. With U as in Lemma 3.3 and with L as in Lemma 3.5 we obtain

$$f_Z(\tilde{x}) \preccurlyeq_u U \preccurlyeq_u L \preccurlyeq_u f_Z(x) \text{ for all } x \in S^*,$$

i.e. no $x \in S^*$ can be a strictly optimal solution of the set optimization problem $\min_{x \in S} f_Z(x)$ and therefore no $x \in S$ can be decision robust strictly efficient for (P) w.r.t. Z. □

For numerical reasons it is not only important that the sets L and U can easily be calculated. Thereby, we assume that convex single-objective optimization problems can easily be solved, as any locally optimal solution is already a globally optimal solution. It is also important that the sets L and U have a simple structure such that they can easily be compared w.r.t. the upper-type less order relation. In case L and U are finite sets this can be done with a pairwise comparison. In case U is not finite, i.e. not just the anti-ideal point, but $m = 2$, then U is just a line segment and $U \preccurlyeq_u L$ can still easily be checked. For $m \geq 3$ such comparisons get already more complicated in case U is not finite, as it is then a subset of a hyperplane. This is even more the case if several points $\bar{p}$ are used in $(P_{\tilde{x},\bar{p}})$ to construct improved outer approximations, as it was done in [28]. In that case the concept of local upper bounds is helpful which is used and explained in detail in [28]. As this is an implementation issue and does not add insights to the main idea of the discarding test above, we limit ourselves here to cases which can easily be implemented.

4 The algorithm and numerical results

In this section, we derive our algorithm based on the proposed discarding test in Theorem 3.6. We also illustrate and discuss the algorithm on several test instances.

4.1 The branch-and-bound algorithm

As already mentioned, the base of our algorithm is a branch-and-bound approach in which we partition the box X into subboxes. Then we try to discard subboxes which do not contain decision robust strictly efficient solutions.

Algorithm 1 Algorithm for Multiobjective Optimizations Problems with Decision Uncertainty

INPUT: $X \in \mathbb{IR}^n$ with $S \subseteq X$, $\hat{Z} \in \mathbb{IR}^n$ with $Z \subseteq \hat{Z}$, $f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^m)$, $Z^* = \{z^1, \ldots, z^p\}$
OUTPUT: $\mathcal{L}_{S,\text{final}}$
1: $\mathcal{L}_W \leftarrow \{X\}, \mathcal{L}_S \leftarrow \emptyset, \mathcal{L}_{S,\text{final}} \leftarrow \emptyset$
2: **if** $\text{mid}(X) \in S$ **then**
3: $\quad \tilde{x} \leftarrow \text{mid}(X)$
4: $\quad$ Compute U with $f_Z(\tilde{x}) \preccurlyeq_u U$, see Lemmas 3.2 and 3.3
5: $\quad$ Store U in $\mathcal{L}_U$
6: **else** $\mathcal{L}_U \leftarrow \emptyset$
7: **end if**
8: **while** $\mathcal{L}_W \neq \emptyset$ **do**
9: $\quad$ Select a box X^* from $\mathcal{L}_W$ and delete it from $\mathcal{L}_W$
10: $\quad$ Bisect X^* perpendicularly to a direction of maximum width $\rightarrow X^1, X^2$
11: $\quad$ **for** $l = 1, 2$ **do**
12: $\quad\quad$ **if** $X^l \cap S = \emptyset$ **then** Discard X^l
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ Compute L such that $L \preccurlyeq_u f_Z(x)$ for all $x \in X^l \cap S$, see Lemma 3.5 and obtain an $\tilde{x} \in X^l \cap S$
15: $\quad\quad\quad$ **if** there is an $U \in \mathcal{L}_U$ with $U \preccurlyeq_u L$ **then** Discard X^l
16: $\quad\quad\quad$ **else**
17: $\quad\quad\quad\quad$ Compute U with $f_Z(\tilde{x}) \preccurlyeq_u U$, see Lemmas 3.2 and 3.3
18: $\quad\quad\quad\quad$ Store U in $\mathcal{L}_U$ and update $\mathcal{L}_U$
19: $\quad\quad\quad\quad$ **if** $\omega(X^l) < \delta$ **then** Store X^l with L in $\mathcal{L}_S$
20: $\quad\quad\quad\quad$ **else** Store X^l in $\mathcal{L}_W$
21: $\quad\quad\quad\quad$ **end if**
22: $\quad\quad\quad$ **end if**
23: $\quad\quad$ **end if**
24: $\quad$ **end for**
25: **end while**
26: **while** $\mathcal{L}_S \neq \emptyset$ **do**
27: $\quad$ Select a box X^* with its lower bound L from $\mathcal{L}_S$ and delete it from $\mathcal{L}_S$
28: $\quad$ **if** there is a $U \in \mathcal{L}_U$ with $U \preccurlyeq_u L$ **then** Discard X^l
29: $\quad$ **else** Store X^l in $\mathcal{L}_{S,\text{final}}$
30: $\quad$ **end if**
31: **end while**

We are working with three lists. The list $\mathcal{L}_W$ is the working list which collects those boxes which are still of interest. The upper bound sets for some feasible points $\tilde{x} \in S$ are collected in the list $\mathcal{L}_U$. The list $\mathcal{L}_S$ collects those boxes which deliver a first cover of the set of decision robust strictly efficient solutions. In the second `while`-loop from line 26 we check again if boxes from $\mathcal{L}_S$ can be discarded now, because the set of upper bound sets $\mathcal{L}_U$ changes during the main `while`-loop (lines 8 to 25). The final solution list is denoted by $\mathcal{L}_{S,\text{final}}$.

For computing upper bound sets feasible points $\tilde{x} \in S^*$ of a current considered box X^* are required. To reduce the numerical effort, the midpoints of the boxes can be used, i.e. $\tilde{x} := \text{mid}(X^*)$ as far as $\tilde{x} \in S$. Another possibility to obtain feasible points is the following: in line 14, see also Lemma 3.5, for each $z \in Z^*$ and for each objective function f_j an optimization problem is solved with feasible set S^*. The minimal solutions can thus be used as feasible points for computing upper bound sets.

To get only one upper bound set per considered box and to make numerical experiments comparable, we use the minimal solution of the first (underestimated) objective function without uncertainties, i.e. with $z^* = 0$.

After adding a new upper bound set to $\mathcal{L}_U$ an update procedure is applied in line 18: If there is a set $U' \in \mathcal{L}_U$ with $U' \preccurlyeq_u U$ for some $U \in \mathcal{L}_U$, all those dominated U are removed from $\mathcal{L}_U$. Moreover, U' is not stored in $\mathcal{L}_U$ if $U \preccurlyeq_u U'$ for one $U \in \mathcal{L}_U$. This reduces the amount of comparisons for checking the conditions in lines 15 and 28.

We stop the algorithm when all boxes are either discarded or put into the solution list, i.e. in case the working list is empty. We put a box X^* to the solution list if its box width is small enough, i.e. for given $\delta > 0$ it holds: $\omega(X^*) < \delta$.

Theorem 4.1 *Let $\mathcal{L}_{S,\text{final}}$ be the output of the algorithm on page 15 and X_E the set of decision robust strictly efficient solutions of (P) w.r.t. Z. Then the following holds:*

(i) $X_E \subseteq \bigcup_{X \in \mathcal{L}_{S,\text{final}}} X \subseteq \bigcup_{X \in \mathcal{L}_S} X$

(ii) For every box $X^* \in \mathcal{L}_{S,\text{final}}$ it holds: $\omega(X^*) < \delta$

Moreover, the algorithm terminates after finitely many subdivisions in line 10.

Proof. Property (i) holds because of Theorem 3.6. In line 19 only boxes X^* with $\omega(X^*) < \delta$ are stored in the solution list $\mathcal{L}_S$. The final elimination in the second `while`-loop of the algorithm just discards some boxes from $\mathcal{L}_S$, but the box width of the boxes does not change. This proves (ii).

The boxes are bisected perpendicularly to a direction of maximum width. Boxes are either discarded or partitioned until the subboxes have a width smaller than δ. Hence, even if no box is discarded, all subboxes have a width smaller than δ after a finite number of subdivisions. □

We would like to mention that our algorithm does not guarantee to find anything like almost decision robust strictly efficient solutions or that with decreasing δ the covering of X_E gets arbitrarily close. For being able to show something like that we would need lower and upper bound sets which get closer to the sets which are compared for smaller boxes. With Remark 3.1 it follows that for smaller boxes X^* the lower bounds for $f(S^* + z^*)$ for one $z^* \in Z^*$ are indeed tighter than for larger boxes. For an upper bound set U one has to calculate concave overestimators $\overline{f}_j$ for each f_j on the box $\{\tilde{x}\} + \hat{Z}$. However, the distance between $\overline{f}_j$ and f_j does not decrease, when the boxes X^* become smaller, as it only depends on the size of $\hat{Z}$ (and the non-concavity of f_j).

4.2 Numerical results

The proposed algorithm has been implemented in Matlab R2018a and uses the toolbox Intlab [30] for interval arithmetic. All experiments have been done on a computer with Intel(R) Core(TM) i5-7400T CPU and 16Gbytes RAM on operating system WINDOWS 10 ENTERPRISE.

As just recently the notion of decision robust strictly efficient solutions has been introduced in multiobjective optimization, there are no test instances from the literature so far. Here, we have introduced some test instances where it is possible to calculate the decision robust strictly efficient solution sets also analytically. This allows to verify our results. We will also discuss the impact of various parameters of the algorithm as for instance the choice and the number p of elements of the set Z^* which are used to calculate the lower bounds in Lemma 3.5.

For all instances we used $\delta = 0.05$. In Tables 1, 2 and 3 the number of subdivisions in line 10, the time and the number of boxes in the final solution list are given for each test instance. Also, visualizations of the partitions of the initial box X after the execution of the algorithm are presented. The light gray boxes are the discarded boxes. All boxes from the final solution list $\mathcal{L}_{S,\text{final}}$ are dark gray. In case of convex constraints, boxes can be discarded because of infeasibility. These boxes are white in the figures.

Moreover, we have compared the impact of using upper bounds based on the anti-ideal point only as in Lemma 3.2 and based on an improved outer approximations as defined in Lemma 3.3.

Test instance 4.1 *This convex test function is inspired by [6]:*

$$f(x) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - 5)^2 + (x_2 - 5)^2 \end{pmatrix}$$

with $S = X = [-5, 5] \times [-5, 5]$ *and* $\hat{Z} = Z = [-0.3, 0.1] \times [-0.3, 0.1]$.

The set of decision robust strictly efficient solutions is a line segment:

$$\mathcal{L} = \{\lambda(0.1, 0.1)^T + (1 - \lambda)(5, 5)^T \mid \lambda \in [0, 1]\}.$$

The sets Z^0 *to* Z^3 *are chosen as*

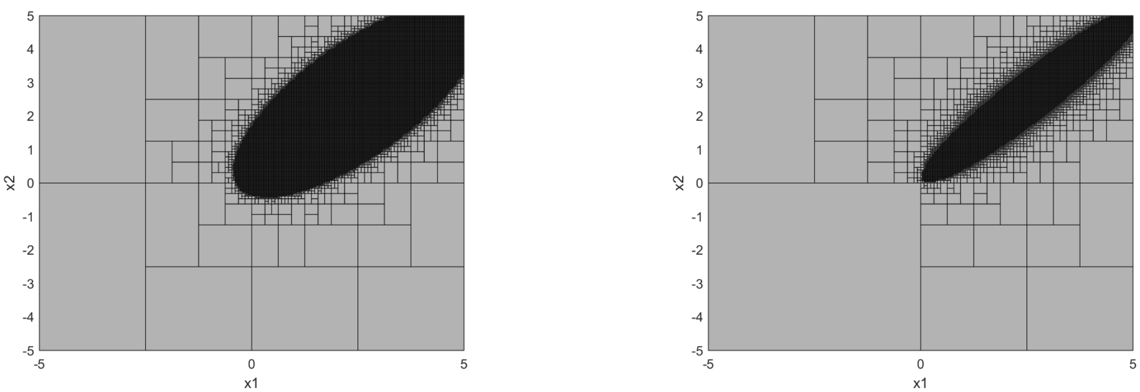
$$\begin{aligned} Z^0 &= \{0\}, \\ Z^1 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.3, 0.1\}\} \cup \{0\}, \\ Z^2 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.3, -0.1, 0.1\}\} \cup \{0\} \text{ and} \\ Z^3 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.3, -0.2, -0.1, 0, 0.1\}\}. \end{aligned}$$

The results of the algorithm are shown in Figure 6 and Table 1.

In Table 1 and Figure 6 it can be seen, that choosing a larger set Z^* and computing the lower bound set U by an improved outer approximation leads to better results. However, the improvement from the case $Z^* = Z^1$ to $Z^* = Z^2$ or $Z^* = Z^3$ is not as significant as the one from $Z^* = Z^0$ to $Z^* = Z^1$. The best choice for Z^* depends on how tight the covering of the set of decision robust strictly efficient solutions should be. Note that the point $\bar{x} = 0$ is not a decision robust strictly efficient solution while $\bar{x}$ is an efficient solution of the corresponding multiobjective optimization problem without uncertainties.

Table 1: Results for Test instance 4.1.

		U by anti-ideal points			U by improved outer approx.		
Z^*	$\lvert Z^*\rvert$	# subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$	# subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$
Z^0	1	12213	**2.1643e+03**	11354	12213	3.0841e+03	11354
Z^1	5	9594	2.9426e+03	8776	7615	2.9225e+03	6686
Z^2	10	9593	4.4985e+03	8776	6714	3.6236e+03	5570
Z^3	25	9593	9.2048e+03	8776	**6175**	6.3523e+03	**4704**



(a) $Z^* = Z^0$, upper bounds by Lemma 3.2 or by Lemma 3.3.

(b) $Z^* = Z^3$, upper bounds by Lemma 3.3.

Figure 6: Partition of the feasible set of Test instance 4.1 after the algorithm.

Test instance 4.2 *This test instance consists of a nonconvex objective function and a circular decision uncertainty set Z:*

$$f(x) = \begin{pmatrix} x_1^2 - x_2^2 \\ x_1/x_2 \end{pmatrix}$$

$$\textit{with } S = X = [-1,2] \times [1,2] \textit{ and } Z = \{(z_1,z_2) \in \mathbb{R}^2 \mid z_1^2 + z_2^2 \leq 0.01\}.$$

A box $\hat{Z}$ which contains Z is $\hat{Z} = [-0.1, 0.1] \times [-0.1, 0.1]$.

The sets Z^0 and Z^1 are chosen as

$$\begin{aligned} Z^0 &= \{0\}, \\ Z^1 &= \{(z_1,z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.1, -0.05, 0, 0.05, 0.1\}\} \cap Z. \end{aligned}$$

For the results of the algorithm see Figure 7 and Table 2.

Test instance 4.3

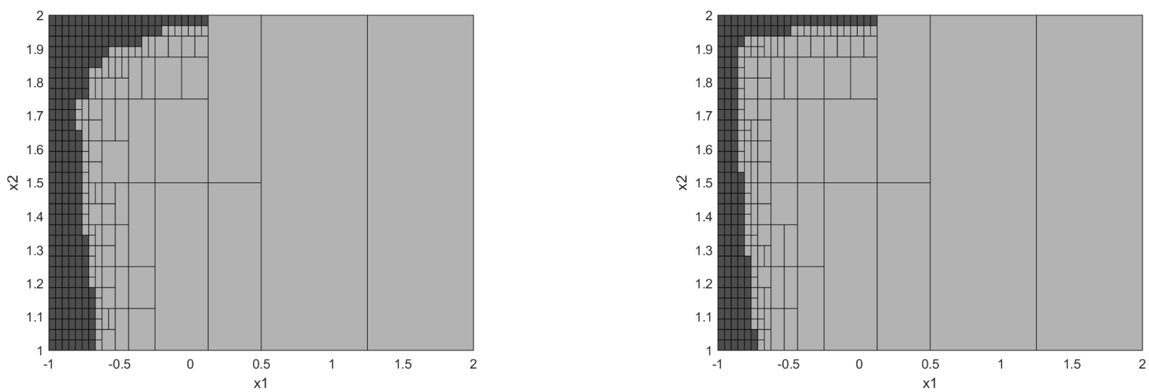
$$f(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix}$$

$$\textit{with } S := \left\{ (x_1,x_2) \in \mathbb{R}^2 \,\middle|\, \begin{array}{ll} x_1^2 + x_2^2 & \leq 0.5 \\ x_1 - x_2 & \leq 0.5 \end{array} \right\} \cap [-1,1] \times [-1,1]$$

$$\textit{and } \hat{Z} = Z = [-0.1, 0.3] \times [-0.1, 0.3]$$

Table 2: Results for Test instance 4.2.

Z^*	$\lvert Z^*\rvert$	U by anti-ideal points # subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$	U by improved outer approx. # subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$
Z^0	1	454	**90**	359	454	160	359
Z^1	13	317	198	224	**264**	201	**154**



(a) $Z^* = Z^1$, upper bounds by Lemma 3.2. (b) $Z^* = Z^1$, upper bounds by Lemma 3.3.

Figure 7: Partition of the feasible set of Test instance 4.2 after the algorithm.

The set of the decision robust strictly efficient solutions is

$$\mathcal{L} = \{(-0.1, -0.1)^T\}.$$

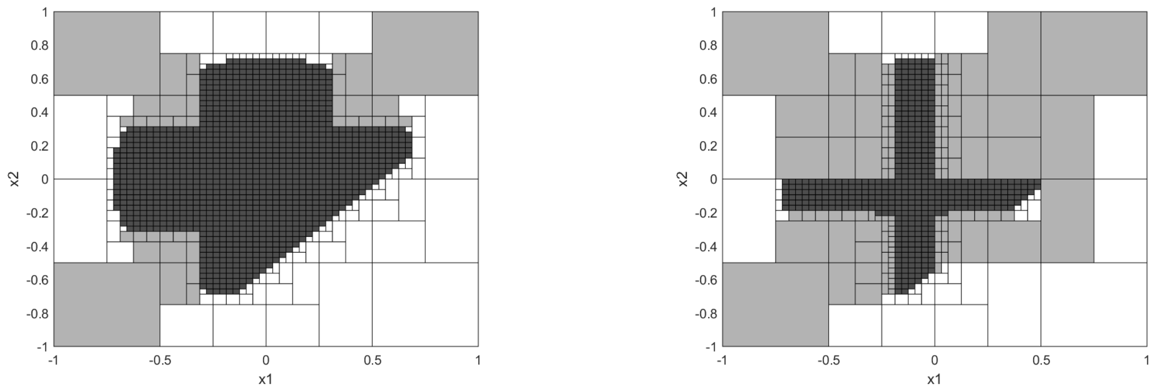
The sets Z^0 *to* Z^3 *are chosen as*

$$\begin{aligned} Z^0 &= \{0\}, \\ Z^1 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.1, 0.3\}\} \cup \{0\}, \\ Z^2 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{0, 0.1, 0.2\}\} \text{ and} \\ Z^3 &= \{(z_1, z_2) \in \mathbb{R}^2 \mid z_1, z_2 \in \{-0.1, 0, 0.1, 0.2, 0.3\}\}. \end{aligned}$$

Note that Z^1 *consists of the vertices of* Z *(and 0) while* Z^2 *contains some interior points of* Z *only. The result of the algorithm is shown in Figure 8 and Table 3.*

Table 3: Results for Test instance 4.3.

Z^*	$\lvert Z^*\rvert$	U by anti-ideal points # subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$	U by improved outer approx. # subdiv.	t [s]	$\lvert\mathcal{L}_{S,\text{final}}\rvert$
Z^0	1	1309	**174**	1171	1309	264	1171
Z^1	5	**623**	183	**453**	**623**	226	**453**
Z^2	9	1011	475	857	1011	553	857
Z^3	25	**623**	694	**453**	**623**	737	**453**



(a) $Z^* = Z^0$, upper bounds by Lemma 3.2 or by Lemma 3.3.

(b) $Z^* = Z^1$, $Z^* = Z^2$ or $Z^* = Z^3$, upper bounds by Lemma 3.2 or by Lemma 3.3.

Figure 8: Partition of the feasible set of Test instance 4.3 after the algorithm.

The results for this test instance, see Table 3, show that the upper bound sets which are obtained from the improved outer approximations do not lead to any improvements. The number of subdivision and boxes in the solution list $\mathcal{L}_{S,\text{final}}$ are the same, which is caused by the simplicity of the objective functions. The computational time is even higher if the upper bound sets are obtained by outer approximations, because more optimization problems have to be solved.

It can be seen that choosing a set Z^* with multiple points improves the results. If the vertices of Z are included in Z^*, i.e in case of Z^1 and Z^3, we obtain the smallest amount of subdivisions and boxes in the solution list. The reason for this is the simple structure of the objective functions again. Therefore for this example, the best choice for Z^* is Z^1 and it is sufficient to use the anti-ideal point of the concave overestimators only. Another thing is that the covering of the decision robust strictly efficient solution $(-0.1, -0.1)^T$ is cross shaped and lies more symmetrically around the real decision robust solution set in case $Z^* \neq \{0\}$.

In all additional numerical experiments similar results were observed. To summarize, choosing a set Z^* with more than one element is a better choice than $Z^* = \{0\}$. On the other side, the cardinality of Z^* does not have to be very large. For simple objective functions the set with some characteristic points at the boundary of Z, e.g. the vertices of $\hat{Z}$ and 0, leads to the overall best results. In general, using outer approximations to obtain upper bound sets improves the results.

5 Conclusions and Outlook

In this work we proposed a branch-and-bound based algorithm for multiobjective optimization problems with decision uncertainty. In case the objective functions are not linear, we need to work with convex underestimators or concave overestimators, or even with both, to make the subproblems numerically tractable. The computational experiments showed that the algorithm is indeed able to discard areas which do not contain any decision robust strictly efficient solution. Nevertheless, the remaining parts of the feasible set can still be large and then other (local) algorithms could be applied afterwards for more exact solutions. Moreover, the results can be improved, i.e. the covering can be tightened by improving the outer approximation used in Lemma 3.3 with techniques from [8] and [25].

We have assumed that the set Z is convex. An adaption also to nonconvex sets is possible. One can replace the set Z within the optimization problems in Subsection 3.2 by the convex hull of Z or by a box, which contains Z. In case Z is a finite (and small) set the problems can also be solved directly by enumeration.

The proposed techniques have been developed for a set optimization problem with a very specific structure and with the upper-type less order relation. It is also of interest, and possible, to adapt the methods for other set order relations and to more general set-valued optimization problems.

6 Acknowledgments

The second author thanks the Carl-Zeiss-Stiftung and the DFG-founded Research Training Group 1567 "Lorentz Force Velocimetry and Lorentz Force Eddy Current Testing" for financial support. The work of the third author is funded by the Deutsche Forschungsgemeinschaft under grant No. EI 821/4.

References

[1] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs: I - Theoretical Advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.

[2] C. Barrico and C.H. Antunes. Robustness analysis in multi-objective optimization using a degree of robustness concept. In *IEEE Congress on Evolutionary Computation. CEC 2006*, pages 1887 –1892. IEEE Computer Society, 2006.

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

[4] A. Ben-Tal and A. Nemirovski. Robust optimization – methodology and applications. *Math. Program.*, 92(3):453–480, 2002.

[5] H.P. Benson. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *J. Global Optim.*, 13(1):1–24, 1998.

[6] T.T. Binh and U. Korn. MOBES: A Multiobjective Evolution Strategy for Constrained Optimization Problems. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 176–182, 1999.

[7] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evol. Comput.*, 14(4):463–494, 2006.

[8] M. Ehrgott, L. Shao, and A. Schöbel. An approximation algorithm for convex multi-objective programming problems. *J. Global Optim.*, 50(3):397–416, 2011.

[9] G. Eichfelder and T. Gerlach. On classes of set optimization problems which are reducible to vector optimization problems and its impact on numerical test instances. *Preprint-Series of the Institute for Mathematics, Technische Universität Ilmenau, Germany*, 2018.

[10] G. Eichfelder, C. Krüger, and A. Schöbel. Decision uncertainty in multiobjective optimization. *J. Global Optim.*, 69(2):485–510, 2017.

[11] J. Fliege and R. Werner. Robust multiobjective optimization & applications in portfolio optimization. *European J. Oper. Res.*, 2013.

[12] M. A. Goberna, V. Jeyakumar, G. Li, and J. Vicente-Pérez. Robust solutions of multiobjective linear semi-infinite programs under constraint data uncertainty. *SIAM J. Optim.*, 24(3):1402–1419, 2014.

[13] J. Jahn. A derivative-free rooted tree method in nonconvex set optimization. to appear.

[14] J. Jahn. Vectorization in set optimization. *J. Optim. Theory Appl.*, 167(3):783–795, 2013.

[15] J. Jahn. A derivative-free descent method in set optimization. *Comput. Optim. Appl.*, 60(2):393–411, 2015.

[16] J. Jahn and T.X.D. Ha. New order relations in set optimization. *J. Optim. Theory Appl.*, 148:209–236, 2011.

[17] A.A. Khan, C. Tammer, and C. Zălinescu. *Set-valued Optimization.* Springer, 2015.

[18] E. Köbis and M.A. Köbis. Treatment of set order relations by means of a nonlinear scalarization functional: a full characterization. *Optimization*, 65(10):1805–1827, 2016.

[19] C. Krüger. *On Minmax Robustness for Multiobjective Optimization with Decision or Parameter Uncertainty.* PhD thesis, Georg-August Universität Göttingen, 2018.

[20] C. Krüger. Peat and pots: Analysis of robust solutions for a biobjective problem in agriculture. *Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen*, 2018.

[21] C. Krüger, F. Castellani, J. Geldermann, and A. Schöbel. Peat and Pots: An application of robust multiobjective optimization to a mixing problem in agriculture. to appear.

[22] D. Kuroiwa. The natural criteria in set-valued optimization. *RIMS Kokyuroku*, 1031:85–90, 1998.

[23] D. Kuroiwa and G. M. Lee. On robust multiobjective optimization. *Vietnam J. Math.*, 40(2&3):305–317, 2012.

[24] A.S. Lewis and C.H.J. Pang. Lipschitz behavior of the robust regularization. *SIAM J. Control Optim.*, 48(5):3080–3105, 2009.

[25] A. Löhne, B. Rudloff, and F. Ulus. Primal and dual approximation algorithms for convex vector optimization problems. *J. Global Optim.*, 60(4):713–736, 2014.

[26] A. Löhne and C. Schrage. An algorithm to solve polyhedral convex set optimization problems. *Optimization*, 62(1):131–141, 2013.

[27] C. D. Maranas and C. A. Floudas. Global minimum potential energy conformations of small molecules. *J. Global Optim.*, 4(2):135–170, 1994.

[28] J. Niebling and G. Eichfelder. A Branch-and-Bound based Algorithm for Nonconvex Multiobjective Optimization. *Preprint-Series of the Institute for Mathematics, Technische Universität Ilmenau, Germany*, 2018.

[29] L. Rodríguez-Marín and M. Sama. (Λ, C)-contingent derivatives of set-valued maps. *J. Math. Anal. Appl.*, 335(2):974–989, 2007.

[30] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, 1999.

[31] M. Schulze Darup and M. Mönnigmann. Improved Automatic Computation of Hessian Matrix Spectral Bounds. *SIAM J. Sci. Comput.*, 38(4):A2068–A2090, 2016.

[32] Y. Zhou-Kangas, K. Miettinen, and K. K. Sindhya. Solving multiobjective optimization problems with decision uncertainty: an interactive approach. *Journal of Business Economics*, 2018.