

A coverage-based Box-Algorithm to compute a representation for optimization problems with three objective functions*

Tobias Kuhn[†] Stefan Ruzika[‡]

November 6, 2014

Abstract

Abstract. A new algorithm for optimization problems with three objective functions is presented which computes a representation for the set of nondominated points. This representation is guaranteed to have a desired coverage error and a bound on the number of iterations needed by the algorithm to meet this coverage error is derived. Since the representation does not necessarily contain nondominated points only, ideas to calculate bounds for the representation error are given. Moreover, the incorporation of domination during the algorithm and other quality measures are discussed.

1 Introduction

This article addresses multiple objective optimization problems which can be concisely stated as

$$\begin{aligned} (\text{MOP}) \quad & \min (f_1(x), \dots, f_p(x)) \\ \text{s. t. } & x \in X \subseteq \mathbb{R}^n \end{aligned}$$

with $f = (f_1, \dots, f_p) : X \rightarrow \mathbb{R}^p$. More precisely, we focus on the case $p = 3$ and our algorithm is applicable to general mathematical programming problems with three objective functions for which the problem

$$\begin{aligned} (P_{\varepsilon_1, \varepsilon_2}^1) \quad & \text{lex min } (f_3(x), f_2(x), f_1(x)) \\ \text{s. t. } & x \in X \\ & \left. \begin{aligned} l_1 &\leq f_1(x) \leq \varepsilon_1 \\ l_2 &\leq f_2(x) \leq \varepsilon_2 \\ l_3 &\leq f_3(x) \leq u_3 \end{aligned} \right\} =: f(x) \in B(l, u)^{(\varepsilon_1, \varepsilon_2, u_3)} \end{aligned}$$

can be solved, e. g. using an appropriate black-box solver.

Since three objective functions are to be optimized simultaneously, the notion of optimality has to be specified.

Definition 1:

A feasible solution $x^* \in X$ of MOP is called *efficient* and its image $f(x^*) \in Y := f(X)$ is called *nondominated*, if there does not exist another solution $\bar{x} \in X$ with $f_i(\bar{x}) \leq f_i(x^*)$, $i = 1, \dots, p$,

*This research has been partially supported by the Federal Ministry of Education and Research Germany, grant DSS Evac Logistic, FKZ 13N12229, and grant FKZ 13N12825.

[†]Fachbereich Mathematik, Technische Universität Kaiserslautern, Kaiserslautern, Germany

[‡]Fachbereich Mathematik/Naturwissenschaften, Mathematisches Institut, Universität Koblenz-Landau, Koblenz, Germany

and $f(\bar{x}) \neq f(x^*)$. The set of all efficient and nondominated solutions are denoted by X_E and Y_N , and they are called the *efficient set* and *nondominated set*, respectively.

The ideal, nadir and anti-ideal point are defined as

$$y_i^I := \min_{x \in X} f_i(x), \quad y_i^N := \max_{x \in X_E} f_i(x) \quad \text{and} \quad y_i^{AI} := \max_{x \in X} f_i(x) \quad \forall i = 1, \dots, p,$$

respectively.

The following notation allows comparisons between two vectors $u, v \in \mathbb{R}^p$ based on the componentwise ordering.

$$\begin{aligned} u \leq v & \quad :\Leftrightarrow \quad u_i \leq v_i \quad \forall i = 1, \dots, p \\ u \leq v & \quad :\Leftrightarrow \quad u \neq v \text{ and } u_i \leq v_i \quad \forall i = 1, \dots, p. \end{aligned}$$

Using the binary relation \leq on \mathbb{R}^p , we define the cone $\mathbb{R}_{\geq}^p := \{x \in \mathbb{R}^p : x \geq 0\}$. \diamond

Our algorithm utilizes cuboids—referred to as *boxes* in the sequel—in \mathbb{R}^3 to bound the nondominated set of the optimization problem at hand. A box $B(l, u)$ can be defined by

$$B(l, u) = l + \mathbb{R}_{\geq}^3 \cap u - \mathbb{R}_{\geq}^3 = \{y \in \mathbb{R}^3 : l \leq y \leq u\},$$

i. e. it is unambiguously described by specifying its “lower, left corner” $l \in \mathbb{R}^3$ and its “upper, right corner” $u \in \mathbb{R}^3$. For some arbitrary norm $\|\cdot\|$ on \mathbb{R}^3 , the metric or distance $\|l - u\|$ is referred to as the *corner point distance* or also *distance of the corner points* of the box $B(l, u)$.

In practice, it is often not easy nor desirable to compute an exact description of Y_N , since this set typically includes a very large or even infinite number of points. Hence, many research articles focus on the computation of a representation of Y_N consisting of a discrete set of points, referred to as *representative points*. A survey of methods for computing a representation of Y_N can be found e. g. in (Ruzika and Wiecek, 2005). A representation Rep is in the broadest sense understood as any substitute for the nondominated set. However, for the sake of fast and accurate decision making, such a representation should come with some quality guarantees ensuring that the nondominated set Y_N is represented appropriately. For discrete representations, the following four quality measures can be used to assess the quality of a representation.

Definition 2:

Let Y_N denote the nondominated set of some multiple objective optimization problem (MOP). Let $Rep \subset \mathbb{R}^p$ denote a representation of the nondominated set.

a) The *coverage error* of the representation is defined as

$$\max_{y \in Y_N} \min_{z \in Rep} \|z - y\|.$$

b) The *uniformity* of the representation is defined as

$$\min_{\substack{y, \hat{y} \in Rep \\ y \neq \hat{y}}} \|y - \hat{y}\|.$$

c) The *cardinality* of the representation is defined as the cardinality of Rep , i. e. $|Rep|$.

d) The *representation error* of the representation is defined as

$$\max_{z \in Rep} \min_{y \in Y_N} \|y - z\|.$$

\diamond

The first three measures were proposed by Sayın (2000) and the representation error was introduced in (Ruzika, 2007). The *coverage error* quantifies how accurate the representation represents the whole nondominated set. The *uniformity* measures the minimal distance between two different

representative points, which quantifies the diversity of the representation. A low *cardinality* could be of interest to the decision maker and if the representation does not only consist of nondominated points, the *representation error* denotes the largest distance of a representative point to its closest nondominated point. Note that these quality measures are conflicting by nature. For instance, the smaller the coverage error, the larger the cardinality of the representation.

Since our algorithm relies on boxes in \mathbb{R}^3 , recently published approaches also using boxes in the context of multiple objective programming shall briefly reviewed in the following.

Laumanns et al. (2006) as well as the follow-up work of Dhaenens et al. (2010), Kirlik and Sayın (2014) and Boland et al. (2014a) do not compute a representation of the nondominated set but aim at finding the exact nondominated set. Their method fixes one coordinate, projects the p -dimensional outcome space onto the remaining $(p - 1)$ -dimensional space and partitions the projection by means of boxes into a grid. Then, this grid is used to search for new nondominated points by solving ε -constraint subproblems (Chankong and Haimes, 1983). The grid is updated adaptively until the nondominated set is found completely. Laumanns et al. (2006) notice that also a heuristic can be derived to get a representation of the nondominated set. In contrast to Laumanns et al. (2006), Dhaenens et al. (2010) apply a partitioning directly in the p -dimensional outcome space and suggest to calculate the true nadir point as described in the article (Ehrgott and Tenfelde-Podehl, 2003) in a first phase which also yields some intermediate solution points. These solution points are used to determine an initial grid in a lower dimensional projected subspace. Later, Dächert and Klamroth (2013) improve the splitting of the boxes used in (Dhaenens et al., 2010) and propose a generic algorithm (not necessarily using ε -constraint problems but also e. g. augmented weighted Tchebycheff problems) which can also find the whole nondominated set for problems with three objectives while the outcome space is decomposed into relevant boxes. Recently, Boland et al. (2014b) proposed another method for finding all nondominated points for integer programs with three objectives. They use so-called L-shapes and rectangles to partition a two-dimensional projected search space. Moreover, they evaluate experimentally the representation quality with the help of the hypervolume indicator if the algorithm is stopped before all nondominated points are found. All these algorithms use only dominance information and information about the calculated points to split boxes and to guide the search. The considered subproblems either have no lower bound in the fixed coordinate or no lower bound at all (besides the ideal point). These methods do not aim at and are not capable for bounding and controlling the coverage error accurately during the running time; a bound on the number of iterations needed for a predefined desired coverage error is not provided.

Besides these methods, the algorithm of Barichard and Hao (2003) which uses evolutionary concepts considers union of boxes containing the nondominated set. Without solving any problem they first bisect the initial box and all resulting boxes according to a selected objective until they have reached the maximal population size. After that, they try to instantiate each box (with an unspecified instantiation procedure) and then remove dominated boxes, which reduces the current population. The procedure is repeated until the number of (not deleted) instantiated boxes has reached the given maximal population size. The execution of this algorithm can be very expensive, since the division of the boxes is made independently (a priori) of the found solutions. Moreover, they do not use the coverage error but the cardinality of the representation as stopping criterion.

Our Contribution

We extend the *Box-Algorithm* proposed by Hamacher et al. (2007) for two objectives to the case of three objectives. Besides dominance information stemming from the ordering relation \leq in \mathbb{R}^3 as well as information about optimality and feasibility of the lexicographic ε -constraint problem, our algorithm chooses iteratively appropriately placed ε -constraints to subdivide the outcome space and to reduce the coverage error. Moreover, also the lower bounds of the considered boxes are modified such that a partition of the remaining unexplored region into non-overlapping boxes is obtained. The aim of our algorithm is not to enumerate all nondominated solutions but to compute a representation of Y_N while reducing the coverage error. Therefore, the upcoming ε -constraint problems in our algorithm cannot be limited to one fixed first coordinate which is minimized, the algorithm has to choose the appropriate first coordinate and thereby the appropriate subdivision

of boxes, i.e. the two ε -constraints, adaptively in each iteration. Since lower bounds are used for the boxes, optimal solutions of the ε -constraint subproblems might be dominated. However, in the end, the unexplored space will be reduced and a subdivision of boxes containing the nondominated set with desired accuracy will be received, for which representative points can be calculated (see completion steps). A straightforward extension of (Hamacher et al., 2007) could use the volume of the boxes as a stopping criterion. This stopping condition can also be implemented in our new algorithm, and the needed analogue results for three objectives are also proven. However, since we experienced that the volume of boxes is not a robust and reliable quality measure, as lower-dimensional boxes would get an empty volume and boxes with two huge sides and one very small side would not be penalized, the focus in this document lies on the coverage error introduced in (Sayın, 2000). Furthermore, it will also be explained how the other quality measures, i.e. cardinality and uniformity, can be treated in the theoretical analysis of the algorithm. Since the new algorithm can produce dominated representative points, the representation error could be greater than 0.

The remainder of this article is organized as follows. The construction of the initial box and subsequently, the update step, i.e. the subdivision of the remaining unexplored search region after a new point has been found, are presented in Section 2. In Section 3, the subdivision is analyzed theoretically. Completion steps for getting a representation as desired are described in Section 4. In Section 5, ideas for bounding the value of the representation error are outlined. Different extensions and improvements for our algorithm are mentioned in Section 6.

2 Initialization and Update Step

The algorithm is initialized with a starting box $B(l^0, u^0)$ with lower bound l^0 and upper bound u^0 . When choosing the initial lower and upper bounds l^0 and u^0 , we require that $Y_N \subseteq B(l^0, u^0)$. The ideal point y^I is an obvious candidate for the lower bound l^0 . For the upper bound u^0 , either an acceptable upper bound from the decision maker, the anti-ideal point y^{AI} , the nadir point y^N computed by the technique described by Ehrgott and Tenfelde-Podehl (2003) or a problem-specific upper bound can be used.

After the initialization and during the execution of the algorithm, every box consists of edges which are parallel to the coordinate axis.

Suppose a box $B(l, u)$ —referred to as the *current box*—is chosen in some iteration due to some *selection rule* (e.g. the box with the biggest corner point distance). It is described in the following how the representation is refined locally, i.e. in the area contained within this current box.

First, the two longest edges of the current box are determined and one ε -constraint problem is solved with ε -constraints bisecting these two longest edges. Depending on the coordinate axis which are parallel to these two longest edges, we refer to this ε -constraint problem either as $(P_{\varepsilon_1, \varepsilon_2}^1)$, $(P_{\varepsilon_1, \varepsilon_3}^2)$, or $(P_{\varepsilon_2, \varepsilon_3}^3)$.

$$\begin{array}{ll} (P_{\varepsilon_1, \varepsilon_3}^2) & \text{lex min } (f_2(x), f_1(x), f_3(x)) \\ & \text{s. t. } x \in X \\ & f(x) \in B(l, u)^{(\varepsilon_1, u_2, \varepsilon_3)} \end{array} \quad \begin{array}{ll} (P_{\varepsilon_2, \varepsilon_3}^3) & \text{lex min } (f_1(x), f_3(x), f_2(x)) \\ & \text{s. t. } x \in X \\ & f(x) \in B(l, u)^{(u_1, \varepsilon_2, \varepsilon_3)} \end{array}$$

Here, the value of ε_i is chosen as

$$\varepsilon_i = l_i + \frac{u_i - l_i}{2}$$

for $i = 1, 2, 3$.

Without loss of generality, we may assume for the subsequent exposition of our ideas that the first subproblem $(P_{\varepsilon_1, \varepsilon_2}^1)$ applies.

Proposition 3: Let z^* be the outcome of an optimal solution of $(P_{\varepsilon_1, \varepsilon_2}^1)$. Then there is no $y \in Y_N \setminus \{z^*\}$ such that

$$y \in B(z^*, u) \cup B \left(l, \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix} \right) \setminus B \left(\begin{pmatrix} l_1 \\ z_2^* \\ z_3^* \end{pmatrix}, \begin{pmatrix} z_1^* \\ \varepsilon_2 \\ z_3^* \end{pmatrix} \right) .$$

Proof:

The set $B(z^*, u)$ is dominated by z^* and the set $B \left(l, \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix} \right) \setminus B \left(\begin{pmatrix} l_1 \\ z_2^* \\ z_3^* \end{pmatrix}, \begin{pmatrix} z_1^* \\ \varepsilon_2 \\ z_3^* \end{pmatrix} \right)$ cannot contain a feasible outcome since this contradicts optimality of z^* for the lexicographic subproblem. \square

Example 4:

We consider an initial box with $l = 0$ and $u = (12, 10, 10)^T$. We solve subproblem $(P_{\varepsilon_1, \varepsilon_2}^1)$ with $\varepsilon_1 = 6$ and $\varepsilon_2 = 5$. Suppose the invoked subproblem has returned the point $z^* = (2, 3, 4)^T$. Then, due to Proposition 3, the two regions

$$B \left(\begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 12 \\ 10 \\ 10 \end{pmatrix} \right) \quad \text{and} \quad B \left(0, \begin{pmatrix} 6 \\ 5 \\ 4 \end{pmatrix} \right) \setminus B \left(\begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix} \right)$$

do not contain unknown nondominated points and can thus be excluded from further consideration. These two sets are depicted in Figure 1 in light and dark gray color, respectively.

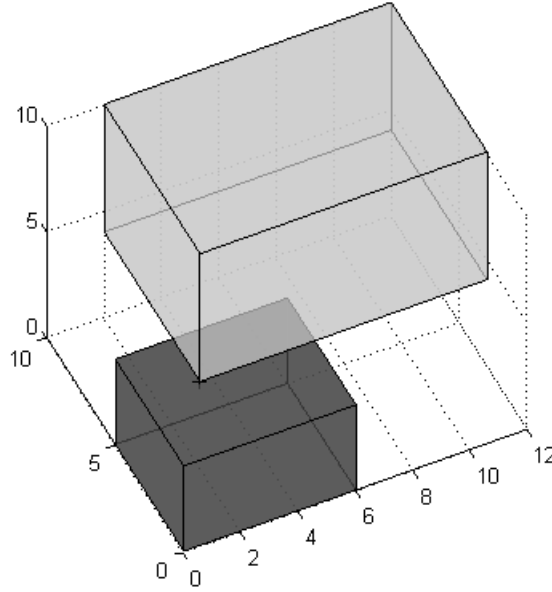


Figure 1: Excluded Regions for Example 4.

\diamond

Excluding the two regions introduced in Proposition 3 from the current box $B(l, u)$, the remaining subset of the current box can be subdivided into new boxes the collection of which contains $Y_N \cap B(l, u)$. The search space is thus reduced. The boxes of the remaining subset of the current box are located in each quarter of the current box (induced by the two ε -constraints).

These different quarters of the current box are denoted by

$$Q_{1,1} := B \left(l, \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ u_3 \end{pmatrix} \right), \quad Q_{1,2} := B \left(\begin{pmatrix} \varepsilon_1 \\ l_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} u_1 \\ \varepsilon_2 \\ u_3 \end{pmatrix} \right),$$

$$Q_{1,3} := B \left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ l_3 \end{pmatrix}, u \right), \quad Q_{1,4} := B \left(\begin{pmatrix} l_1 \\ \varepsilon_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} \varepsilon_1 \\ u_2 \\ u_3 \end{pmatrix} \right).$$

where the first subscript index refers to the number of problem ($P_{\varepsilon_1, \varepsilon_2}^1$) and the second simply enumerates the quarters.

The subdivision of $B(l, u)$ into boxes is not unique. Although other subdivisions are eligible, the following is used (note that the subsequent analysis applies to alternative subdivisions as long as each box is completely contained in a quarter and does not overlap with another box under consideration). The newly obtained boxes are

$$B^{1,1} := B \left(\begin{pmatrix} l_1 \\ z_2^* \\ z_3^* \end{pmatrix}, \begin{pmatrix} z_1^* \\ \varepsilon_2 \\ u_3 \end{pmatrix} \right), \quad B^{1,2} := B \left(\begin{pmatrix} l_1 \\ l_2 \\ z_3^* \end{pmatrix}, \begin{pmatrix} \varepsilon_1 \\ z_2^* \\ u_3 \end{pmatrix} \right)$$

for $Q_{1,1}$,

$$B^{1,3} := B \left(\begin{pmatrix} \varepsilon_1 \\ l_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} u_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix} \right), \quad B^{1,4} := B \left(\begin{pmatrix} \varepsilon_1 \\ l_2 \\ z_3^* \end{pmatrix}, \begin{pmatrix} u_1 \\ z_2^* \\ u_3 \end{pmatrix} \right)$$

for $Q_{1,2}$,

$$B^{1,5} := B \left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} u_1 \\ u_2 \\ z_3^* \end{pmatrix} \right)$$

for $Q_{1,3}$, and

$$B^{1,6} := B \left(\begin{pmatrix} l_1 \\ \varepsilon_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} \varepsilon_1 \\ u_2 \\ z_3^* \end{pmatrix} \right), \quad B^{1,7} := B \left(\begin{pmatrix} l_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix}, \begin{pmatrix} z_1^* \\ u_2 \\ u_3 \end{pmatrix} \right)$$

for $Q_{1,4}$.

The corresponding quarters $Q_{2,i}$ and $Q_{3,i}$, $i = 1, \dots, 4$, and subdivisions $B^{2,j}$ and $B^{3,j}$, $j = 1, \dots, 7$, for the subproblems ($P_{\varepsilon_1, \varepsilon_3}^2$) and ($P_{\varepsilon_2, \varepsilon_3}^3$), respectively, are defined analogously.

In Figure 2, the unexplored region and the subdivision for Example 4 is depicted.

Remark 5:

- If the point z^* is located on the boundary of the current box, the subdivision should be further adjusted such that the resulting boxes do not overlap. Since these modifications are in principle trivial but would lengthen the exposition of the algorithm, the focus here is on the case that z^* is in the interior of the current box.

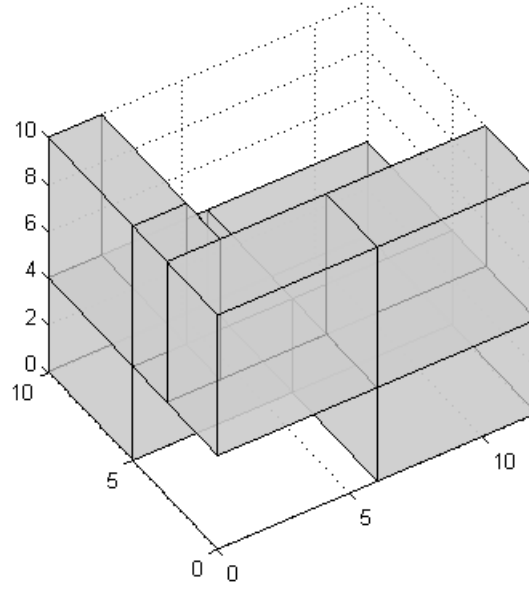


Figure 2: Unexplored Region

- If the lower bound constraints in the considered subproblems are omitted, it is not guaranteed to find a solution lying in the desired box, since for example the point $(\varepsilon_1, l_2, l_3)^T$ in box $B^{1,3}$ dominates every point in the boxes $B^{1,4}$ and $B^{1,5}$ w.r.t. the scalarization problem. Therefore, the additional lower bound constraints for the objectives are necessary to guarantee that the obtained solution is located in the desired box. Since the considered point $(\varepsilon_1, l_2, l_3)^T$ dominates every point of the current box with first coordinate greater than or equal to ε_1 w.r.t. a considered subproblem in the boxes $B^{1,3}$, $B^{1,4}$ or $B^{1,5}$, also every other partition in that unexplored region of the current box into boxes can lead to such difficulties. Similar issues can be observed for the remaining unexplored region of the current box, i.e. with first coordinate less than or equal to ε_1 .

◇

3 Properties of the Subdivision

After problem $(P_{\varepsilon_1, \varepsilon_2}^1)$ was solved for the current box $B(l, u)$, the box is subdivided into the seven new boxes $B^{1,1}$ to $B^{1,7}$ defined as above for the quarters $Q_{1,1}$ to $Q_{1,4}$. This step is repeated until the corner point distances of all boxes have reached a value of at most δ^C . Nevertheless, the computed representative points are not guaranteed to be nondominated. Yet, due to feasibility and the structure of the subproblem $(P_{\varepsilon_1, \varepsilon_2}^1)$, the following result is obtained:

Lemma 6: $Y_N \cap B(l, u) \subseteq \bigcup_{i=1}^7 B^{1,i}$

Proof:

Follows directly from Proposition 3. □

An analogous result as in (Hamacher et al., 2007) about the reduction of the volume of the boxes after an update step can be proven:

Observation 7:

If a box $B^0 := B(l, u)$, after solving the corresponding subproblem, is divided into the seven new boxes, each box $B^{1,i}$, $i = 1, \dots, 7$ fulfills

$$\text{Vol}(B^{1,i}) \leq \frac{1}{4} \cdot \text{Vol}(B^0)$$

where $\text{Vol}(B^{1,i})$ measures the three-dimensional volume of box $B^{1,i}$. In particular, for three quarters two pairs of boxes can be found, namely the two lying in the same quarter, where the combined volume fulfills this formula. \diamond

The next lemma is about the total volume of the boxes $B^{1,1}$ to $B^{1,7}$ in view of the volume of the box $B^0 := B(l, u)$:

Lemma 8:

$$\sum_{i=1}^7 \text{Vol}(B^{1,i}) \leq \frac{3}{4} \cdot \text{Vol}(B^0)$$

Proof:

The volume of the boxes $B^{lex} := B\left(l, \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix}\right)$ and $B^{dom} := B(z^*, u)$ are cut off fulfilling

$$\begin{aligned} \text{Vol}(B^{lex}) + \text{Vol}(B^{dom}) &\geq \text{Vol}(B^{lex}) + \text{Vol}\left(B\left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix}, u\right)\right) \\ &= \text{Vol}\left(B^{lex} + \begin{pmatrix} \varepsilon_1 - l_1 \\ \varepsilon_2 - l_2 \\ 0 \end{pmatrix}\right) + \text{Vol}\left(B\left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix}, u\right)\right) \\ &= \text{Vol}\left(B\left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ l_3 \end{pmatrix}, \begin{pmatrix} u_1 \\ u_2 \\ z_3^* \end{pmatrix}\right)\right) + \text{Vol}\left(B\left(\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ z_3^* \end{pmatrix}, u\right)\right) \\ &= \text{Vol}(Q_{1,3}) \end{aligned}$$

where $B^{lex} + \begin{pmatrix} \varepsilon_1 - l_1 \\ \varepsilon_2 - l_2 \\ 0 \end{pmatrix}$ denotes the box after shifting B^{lex} in direction $\begin{pmatrix} \varepsilon_1 - l_1 \\ \varepsilon_2 - l_2 \\ 0 \end{pmatrix}$.

Hence, we get

$$\sum_{i=1}^7 \text{Vol}(B^{1,i}) = \text{Vol}(B^0) - (\text{Vol}(B^{lex}) + \text{Vol}(B^{dom})) \leq \text{Vol}(B^0) - \text{Vol}(Q_{1,3}) = \frac{3}{4} \cdot \text{Vol}(B^0). \quad \square$$

The Box-Algorithm for three objectives is stated as a pseudo-code description in Algorithm 1.

Algorithm 1 A posteriori algorithm (for three objectives, coverage)

Input: A multiple objective optimization problem with three objectives, $\delta^C > 0$

Output: A subdivision \mathcal{B} fulfilling Theorem 9 and an incomplete representation Rep (see completion steps for completion)

- 1: $S \leftarrow \emptyset$
- 2: $Rep \leftarrow \emptyset$
- 3: $\mathcal{B} \leftarrow \emptyset$
- 4: Compute the starting box $B(l^0, u^0)$
- 5: **if** $\|l^0 - u^0\|_\infty > \delta^C$ **then**
- 6: $S \leftarrow \{B(l^0, u^0)\}$
- 7: **else**
- 8: $\mathcal{B} \leftarrow \{B(l^0, u^0)\}$

Algorithm 1 A posteriori algorithm (for three objectives, coverage) – continued

```

9: while  $S \neq \emptyset$  do      // stopping criterion
10:   Choose the box  $B(y^1, y^2) \in S$  with maximal distance  $\|y^1 - y^2\|_\infty$       // selection rule
11:    $S \leftarrow S \setminus \{B(y^1, y^2)\}$ 
12:    $l \leftarrow y^1, u \leftarrow y^2$ 
13:   if  $\min\{u_1 - l_1, u_2 - l_2\} \geq u_3 - l_3$  then
14:     Solve  $(P_{\varepsilon_1, \varepsilon_2}^1)$  with  $\varepsilon_i = l_i + \frac{u_i - l_i}{2}$ ,  $i = 1, 2$ , and obtain optimal solution  $z^*$ 
15:      $B^i := B(y_l^{B^i}, y_u^{B^i}) := B^{1,i} \forall i = 1, \dots, 7$ 
16:      $Q_i := B(y_l^{Q_i}, y_u^{Q_i}) := Q_{1,i} \forall i = 1, \dots, 4$ 
17:   else if  $\min\{u_1 - l_1, u_3 - l_3\} \geq u_2 - l_2$  then
18:     Solve  $(P_{\varepsilon_1, \varepsilon_3}^2)$  with  $\varepsilon_i = l_i + \frac{u_i - l_i}{2}$ ,  $i = 1, 3$ , and obtain optimal solution  $z^*$ 
19:      $B^i := B(y_l^{B^i}, y_u^{B^i}) := B^{2,i} \forall i = 1, \dots, 7$ 
20:      $Q_i := B(y_l^{Q_i}, y_u^{Q_i}) := Q_{2,i} \forall i = 1, \dots, 4$ 
21:   else      //  $\min\{u_2 - l_2, u_3 - l_3\} \geq u_1 - l_1$ 
22:     Solve  $(P_{\varepsilon_2, \varepsilon_3}^3)$  with  $\varepsilon_i = l_i + \frac{u_i - l_i}{2}$ ,  $i = 2, 3$ , and obtain optimal solution  $z^*$ 
23:      $B^i := B(y_l^{B^i}, y_u^{B^i}) := B^{3,i} \forall i = 1, \dots, 7$ 
24:      $Q_i := B(y_l^{Q_i}, y_u^{Q_i}) := Q_{3,i} \forall i = 1, \dots, 4$ 
25:   if  $z^* \neq \text{NULL}$  then      //  $z^* \in Y$ 
26:      $\text{Rep} \leftarrow \text{Rep} \cup \{z^*\}$ 
27:     for  $i \in \{1, \dots, 7\}$  do
28:       if  $\|y_l^{B^i} - y_u^{B^i}\|_\infty > \delta^C$  then
29:          $S \leftarrow S \cup \{B^i\}$ 
30:       else if  $B^i \neq \emptyset$  then
31:          $\mathcal{B} \leftarrow \mathcal{B} \cup \{B^i\}$ 
32:   else      //  $z^* = \text{NULL}$ 
33:     for  $i \in \{2, 3, 4\}$  do
34:       if  $\|y_l^{Q_i} - y_u^{Q_i}\|_\infty > \delta^C$  then
35:          $S \leftarrow S \cup \{Q_i\}$ 
36:       else if  $Q_i \neq \emptyset$  then
37:          $\mathcal{B} \leftarrow \mathcal{B} \cup \{Q_i\}$ 

```

Finiteness, a bound on the complexity as well as correctness of the algorithm are proven in the following.

Theorem 9: *Algorithm 1 terminates in finitely many steps. It outputs a collection of boxes containing all nondominated points. Each of these boxes can be equipped with a representative point. The representation has a coverage error of at most δ^C (w. r. t. $\|\cdot\|_\infty$). More precisely, the algorithm performs at most $\mathcal{O}\left(\left(\frac{L}{\delta^C}\right)^{2 \cdot \log_2(7)}\right)$ many iterations, where L is the distance of the corner points of the initial box $B(l^0, u^0)$, i. e. $L := \|l^0 - u^0\|_\infty$.*

Proof:

Due to the construction of the boxes, a subdivision is obtained which contains Y_N at each state of the algorithm (see Lemma 6). The stopping criterion of the algorithm guarantees that the coverage error of the representation is at most δ^C if the algorithm terminates. Thus, it suffices to show the termination of the algorithm.

The functioning of the algorithm can be represented by a tree, where each node corresponds to a box in some iteration. More precisely, the root node of the tree corresponds to the initial box and after the first iteration (independently of the subproblem considered) this box is substituted by at most seven new boxes which will be then represented by at most seven child nodes of the father node. Finiteness of the algorithm corresponds to a bounded height of the corresponding tree which will be shown in the following.

Let $B(y^1, y^2)$ be a box corresponding to some node of the tree in an arbitrary but odd level greater than or equal to 3.¹ Let $B(w^1, w^2)$ and $B(v^1, v^2)$ be the boxes corresponding to its father and grandfather nodes, respectively. Consider now the iteration in which the box corresponding to the grandfather node is processed. Without loss of generality let problem $(P_{\varepsilon_1, \varepsilon_2}^1)$ be called, i. e.

$$v_3^2 - v_3^1 \leq \min_{i=1,2} \{v_i^2 - v_i^1\} \leq \max_{i=1,2,3} \{v_i^2 - v_i^1\} = \|v^1 - v^2\|_\infty,$$

resulting in boxes including the box $B(w^1, w^2)$. Due to the special ε -constraints in the subproblem and the definition of the boxes, it holds

$$\max_{i=1,2} \{w_i^2 - w_i^1\} \leq \frac{\|v^1 - v^2\|_\infty}{2} \quad (1)$$

and still $w_3^2 - w_3^1 \leq \|v^1 - v^2\|_\infty$.

Claim: $\|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$

Proof:

Case 1: $w_3^2 - w_3^1 \leq \frac{\|v^1 - v^2\|_\infty}{2} \implies \|y^1 - y^2\|_\infty \leq \|w^1 - w^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$.

Case 2: $w_3^2 - w_3^1 > \frac{\|v^1 - v^2\|_\infty}{2} \implies \frac{\|v^1 - v^2\|_\infty}{2} < w_3^2 - w_3^1 \leq \|v^1 - v^2\|_\infty \xrightarrow{(1)} \text{Subproblem } (P_{\varepsilon_1, \varepsilon_3}^2) \text{ or } (P_{\varepsilon_2, \varepsilon_3}^3) \text{ was called when the box } B(w^1, w^2) \text{ was considered.} \implies \text{Due to the special } \varepsilon\text{-constraints it follows } \|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2} \text{ and the claim is proven.} \quad \diamond$

This means that an arbitrary box $B(y^1, y^2)$ corresponding to a node in level $2k - 1$ ($k \in \mathbb{N}$) fulfills

$$\|y^1 - y^2\|_\infty \leq \frac{L}{2^{k-1}}.$$

Hence, if $k \geq \log_2 \left(\frac{L}{\delta^C} \right) + 1$ then $\|y^1 - y^2\|_\infty \leq \delta^C$ is induced and therefore each box at level $2 \cdot \left\lceil \log_2 \left(\frac{L}{\delta^C} \right) \right\rceil + 1$, fulfills the stopping criterion which shows boundedness of the height of the tree.

Moreover, the tree can have at most

$$\sum_{i=0}^{2 \cdot \left\lceil \log_2 \left(\frac{L}{\delta^C} \right) \right\rceil} 7^i = \frac{7^{2 \cdot \left\lceil \log_2 \left(\frac{L}{\delta^C} \right) \right\rceil + 1} - 1}{6} \in \mathcal{O} \left(\left(\frac{L}{\delta^C} \right)^{2 \cdot \log_2(7)} \right)$$

nodes and, correspondingly, the number of iterations of the algorithm is bounded by the same term. \square

If a representation with coverage error w.r. t. an arbitrary p -norm $\|\cdot\|_p$, $1 \leq p \leq \infty$ is desired, Algorithm 1 can still be used to obtain such a representation. However, δ^C has to be chosen sufficiently small. A rule how δ^C can be chosen, is induced by the following (well-known) lemma.

Lemma 10: *Let $1 \leq p < \infty$ and $n \in \mathbb{N}$ then it holds:*

$$\|x\|_p \leq n^{\frac{1}{p}} \cdot \|x\|_\infty \quad \text{for all } x \in \mathbb{R}^n$$

In particular, if $\|x\|_\infty \leq \delta^C$, then $\|x\|_p \leq n^{\frac{1}{p}} \cdot \delta^C$.

Proof:

Let $i_0 := \arg \max_{i=1, \dots, n} |x_i|$, then it follows

$$\|x\|_p = \left(\sum_{i=1}^n \underbrace{|x_i|^p}_{\leq |x_{i_0}|^p} \right)^{\frac{1}{p}} \stackrel{\text{mon. incr.}}{\leq} (n \cdot |x_{i_0}|^p)^{\frac{1}{p}} = n^{\frac{1}{p}} \cdot |x_{i_0}| = n^{\frac{1}{p}} \cdot \|x\|_\infty \quad \square$$

¹The level of the root node is 1, its children are on level 2 and so on.

4 Completion Steps

After termination of the algorithm, a subdivision of the nondominated set in terms of a collection of boxes is obtained. There does not necessarily need to exist a representative point in each box of the subdivision. If the representative points computed do not suffice the needs of the decision maker, four possibilities to repair this issue are suggested:

1. For each box $B(l, u)$ of the subdivision an additional subproblem, e. g. $(P_{\varepsilon_1, \varepsilon_2}^1)$ with $\varepsilon_1 := u_1$ and $\varepsilon_2 := u_2$, is solved such that either this box can be deleted due to infeasibility or a new representative point of this box is identified. The enlarged set Rep is then outputted to the decision maker.
2. The subdivision is presented to the decision maker who will choose some box of interest $B(l, u)$ and only one additional subproblem corresponding to the chosen box with $\varepsilon_1 := u_1$ and $\varepsilon_2 := u_2$ is solved. Additionally, the boxes where a part of them dominates the chosen box or the outcome found by the subproblem could be considered.
3. The decision maker is asked for one or several points of interest after the initial box is presented. Then, all distances from the desired points to these boxes are calculated and the boxes with minimal distance or all with k -minimal distances (i. e. the minimal, second minimal, ..., k -th minimal) to each desired point are saved and for each such box $B(l, u)$ only one additional subproblem is solved with $\varepsilon_1 := u_1$ and $\varepsilon_2 := u_2$.
4. The decision maker is questioned for a desired number of points, some filtering procedure for reducing the set of boxes to a subset with desired number guaranteeing some quality feature is applied (see e.g. Steuer and Harris (1980)), for each of these boxes a representative point is calculated and presented to the decision maker.

If the first completion step is used, a representation with accuracy δ^C is obtained. The second, third and fourth completion steps are better suited in an interactive variant of our algorithm.

Definition 11:

After Algorithm 1 has terminated and the first completion step was applied, the output Rep is then called *complete representation*. \diamond

Let Rep be a complete representation and let Rep_N denote all points of Rep which are not dominated by any other point in this set. Obviously, deleting points could change the coverage error of the representation. Yet, the following quality guarantee for the set Rep_N can be derived.

Corollary 12: *Let Rep be a complete representation with coverage error less than or equal to δ^C (w. r. t. $\|\cdot\|_\infty$) for the considered problem. Then it is*

$$Y_N \subseteq (Rep_N - (\delta^C, \delta^C, \delta^C)^T) + \mathbb{R}_{\geq}^3.$$

Proof:

Let $y \in Y_N$. Due to the coverage property, there exists $z \in Rep$ with $\|y - z\|_\infty \leq \delta^C$ which implies

$$z - (\delta^C, \delta^C, \delta^C)^T \leq y.$$

If $z \notin Rep_N$, then there exists $\hat{z} \in Rep_N$ with $\hat{z} \leq z$ fulfilling

$$\hat{z} - (\delta^C, \delta^C, \delta^C)^T \leq y. \quad \square$$

Remark 13:

Clearly, the first completion step could also be incorporated in the presented algorithm, namely if the boxes are not added to the list \mathcal{B} of processed boxes when they have reached the desired precision but after they are investigated the next time with a subproblem described in the first completion step. \diamond

5 Representation Error

It is noted above that the resulting representation could contain dominated points. Therefore, the representation error

$$\max_{z \in Rep} \min_{y \in Y_N} \|z - y\|$$

(for $\|\cdot\| := \|\cdot\|_p$ for some $1 \leq p \leq \infty$) can be positive. In this section we are interested in how the representation error can be approximated or bounded from above for a given complete representation Rep with coverage error less than or equal to δ^C (w.r. t. some norm $\|\cdot\|$).

Definition 14:

Let \mathcal{B} be the last subdivision (covering the whole set Y_N) obtained from Algorithm 1. Let $z \in Rep$. Then we define $\mathcal{B}^z := \{B \in \mathcal{B} : B \cap (z - \mathbb{R}_{\geq}^3) \neq \emptyset \wedge z \notin \text{int}(B)\}$, where $\text{int}(B)$ denotes the interior of a box B . If $\mathcal{B}^z \neq \emptyset$ we call z a *critical representative point*. \diamond

A critical representative point is a candidate for a dominated point since the expression

$$B \cap (z - \mathbb{R}_{\geq}^3) \neq \emptyset$$

for some box $B \in \mathcal{B}$ is a necessary condition for $Y_N \cap (z - \mathbb{R}_{\geq}^3) \neq \emptyset$. Note that there is no need to check the box containing z in its interior, since in this case z was found in the completion step and z is then locally nondominated w.r. t. this box.

Proposition 15:

$$\max_{z \in Rep} \min_{y \in Y_N} \|z - y\| \leq \min \left\{ \max_{z \in Rep} \max_{B(l,u) \in \mathcal{B}^z} \|l - z\|, \max_{z \in Rep} \max_{\hat{z} \in B_{\delta^C}^{Rep}(z - \mathbb{R}_{\geq}^3)} \|z - \hat{z}\| + \delta^C \right\}$$

where $B_{\delta^C}^{Rep}(z - \mathbb{R}_{\geq}^3) := \{y \in Rep : \exists \tilde{y} \in z - \mathbb{R}_{\geq}^3 \wedge \|y - \tilde{y}\| \leq \delta^C\}$ and $\max_{B(l,u) \in \mathcal{B}^z} \|l - z\|$ returns the value 0 if $\mathcal{B}^z = \emptyset$.

Proof:

Let $z \in Rep$ be an arbitrary representative point. Then we immediately get

$$\min_{y \in Y_N} \|z - y\| \leq \max_{B(l,u) \in \mathcal{B}^z} \|l - z\|,$$

since for a box $B(l, u)$ the distance $\|l - z\|$ is an upper bound for the distance of any point contained in $B(l, u)$ and dominating z . Hence, the first upper bound is proven.

We define $y^* := \arg \min_{y \in Y_N \cap (z - \mathbb{R}_{\geq}^3)} \|z - y\|$ (assuming existence) and $z^{y^*} := \arg \min_{\tilde{z} \in Rep} \|\tilde{z} - y^*\|$, then it follows

$$\begin{aligned} \min_{y \in Y_N} \|z - y\| &\leq \|z - y^*\| = \|z - y^* + z^{y^*} - z^{y^*}\| \\ &\leq \|z - z^{y^*}\| + \|y^* - z^{y^*}\| \leq \|z - z^{y^*}\| + \delta^C \\ &\leq \max_{\hat{z} \in B_{\delta^C}^{Rep}(z - \mathbb{R}_{\geq}^3)} \|z - \hat{z}\| + \delta^C. \end{aligned} \quad \square$$

Remark 16:

- For $\|\cdot\| = \|\cdot\|_{\infty}$ the second upper bound in Proposition 15 can be substituted with

$$\max_{z \in Rep} \max_{\hat{z} \in Rep \cap (z + (\delta^C, \delta^C, \delta^C)^T - \mathbb{R}_{\geq}^3)} \|z - \hat{z}\| + \delta^C.$$

- Another method to approximate the representation error is to use for example Benson's method (see Ehrgott (2005)) for all critical representative points, which returns some non-dominated point dominating the current critical point (if existent). The maximum over all distances between the critical points and the corresponding found nondominated points, will also serve as an upper bound to the representation error.

- Moreover, the representation error could be forced to be zero, if some filtering procedure is applied after the calculation of the representation, which identifies all dominated points and deletes them afterwards. This could also be done for example with the help of Benson's method which will additionally give a replacement point dominating the deleted point. However, this filtering technique could destroy the accuracy since points representing a box having a "nondominating part" could be deleted, but, taking replacement points into account, a result similar to Corollary 12 could still be proven.

◇

6 Extensions of the Box-Algorithm

The basic concept of the Box-Algorithm can be equipped with several improvements and extensions which may be reasonable in appropriate settings and circumstances. The following proposals are thus facultative.

If a discrete optimization problem (MOP) with three integer-valued objective functions is considered, in each iteration the unexplored region can be shrunk even further using the integrality property of feasible solutions and their images.

6.1 Incorporating Domination during the Algorithm

The Box-Algorithm uses the concept of domination only locally: after an optimal solution of the lexicographic ε -constraint subproblem was found, the part of the current box being dominated by the image of this solution is neglected from further consideration. However, this solution also dominates points outside the current box. This observation can help to reduce the number of boxes significantly without losing performance.

Hence, the step in the algorithm is considered where a new (feasible) outcome z^* in the currently considered box is found (cf. lines 14, 18 and 22 in Algorithm 1). It is natural to use this solution to define the notion of dominated boxes (see also Barichard and Hao (2003)).

Definition 17:

In an arbitrary iteration in Algorithm 1, after a solution z^* in some line 14, 18 or 22 was found, a box $B(l, u) \in \mathcal{B} \cup S$ is called *dominated* by z^* if $z^* \leq l$. ◇

Hence, a dominated box cannot contain points of interest since all possible points in the box are dominated by the found solution z^* . Then, right after the solution z^* was found, before line 26, we execute Algorithm 2 as a subroutine which considers the lists S of unexplored boxes and \mathcal{B} of finished boxes and deletes all boxes dominated by z^* . Additionally, before adding a new box B to S or \mathcal{B} in lines 29, 31, 35 and 37, Algorithm 3 is executed as a subroutine which first investigates the set Rep and checks if the new box is already dominated.

Algorithm 2 Deletion of dominated boxes - Part 1

Input: A solution point z^*

```

1: for  $B(l, u) \in \mathcal{B} \cup S$  do
2:   if  $z^* \leq l$  then
3:     Delete  $B(l, u)$ 
```

Algorithm 3 Deletion of dominated boxes - Part 2

Input: A box $B(l, u)$

```

1: for  $z \in Rep$  do
2:   if  $z^* \leq l$  then
3:     Delete  $B(l, u)$ 
```

On the other hand, during the algorithm, also partially dominated boxes could be obtained.

Definition 18:

In an arbitrary iteration in Algorithm 1, after a solution z^* in some line 14, 18 or 22 was found, a box $B(l, u) \in \mathcal{B} \cup S$ is called *partially dominated* by z^* if $z^* \not\leq l$ but $z^* \leq u$. \diamond

In this case a specific subset of a partially dominated box $B(l, u)$ can be deleted, namely the box

$$B(z^{*,proj} := (\max\{l_1, z_1^*\}, \max\{l_2, z_2^*\}, \max\{l_3, z_3^*\})^T, u)$$

where the left corner point $z^{*,proj}$ is called the $B(l, u)$ -*projection* of z^* . For practical reasons, to avoid cutting off only very small portions, some threshold for partial dominance (e.g. 20 %) can be defined for which it is worth to delete this portion from further considerations. Partially dominance is incorporated in Algorithm 1 analogously to domination with appropriately modified subroutines.

6.2 Other Quality Measures

Another quality measure is the previously mentioned concept of uniformity. The Box-Algorithm can be modified to address this measure as well. To this end, let $0 < \delta^U \leq \delta^C$ be given. The aim is a representation Rep fulfilling

$$\min_{\substack{y, \hat{y} \in Rep \\ y \neq \hat{y}}} \|y - \hat{y}\|_\infty \geq \delta^U. \quad (2)$$

It is assumed that $\delta^U \leq \delta^C$, since otherwise the coverage property could be violated. Again, the step in the algorithm is considered in which a new (feasible) outcome z^* is obtained (cf. lines 14, 18 and 22 in Algorithm 1). Due to (2) the following region should be deleted from further investigation

$$\{z \in \mathbb{R}^3 : \|z - z^*\|_\infty \leq \delta^U\} = B\left(\begin{pmatrix} z_1^* - \delta^U \\ z_2^* - \delta^U \\ z_3^* - \delta^U \end{pmatrix}, \begin{pmatrix} z_1^* + \delta^U \\ z_2^* + \delta^U \\ z_3^* + \delta^U \end{pmatrix}\right). \quad (3)$$

This box has now to be incorporated in the current update step in a straight forward way which will result in another subdivision and, furthermore, the lists of all unexplored boxes S and finished boxes \mathcal{B} have to be investigated and the boxes $B(l, u)$ overlapping with box (3) have to be updated appropriately.

If the quality measure “cardinality” should be incorporated, the algorithm can be aborted if the desired cardinality is reached, i.e. an additional stopping criterion corresponding to the desired cardinality has to be added to line 9 in Algorithm 1. Obviously, this could violate the desired coverage property since the different quality measures tend to be contradicting. In this case, however, the coverage error of the currently obtained representation can be computed by looking at the last obtained subdivision.

7 Conclusion and Future Research

A new algorithm for optimization problems with three objective functions was introduced. This algorithm generates a collection of cuboids comprising the nondominated set. At termination, this collection together with some computed points is guaranteed to meet some specified coverage property. Besides the derivation of some properties, an iteration bound to reach the desired coverage error, bounds for the representation error and ideas for possible extensions are presented.

A selection rule not with respect to distance values but with respect to some dominance relation which is able to combine the coverage property with the quality guarantee from Corollary 12 is under current research. Another open question addresses the choice of a more suitable subproblem:

this subproblem should “check” for a given box $B(l, u)$ if $B(l, u) \cap Y_N \neq \emptyset$ and if so, such a point in this intersection should be computed. In other words, it is desirable to solve the following problem efficiently

$$\begin{aligned} (\hat{P}_{\varepsilon_1, \varepsilon_2}^1) \quad & \text{lex min } (f_3(x), f_2(x), f_1(x)) \\ \text{s. t. } & x \in X_E \\ & f(x) \in B(l, u)^{(\varepsilon_1, \varepsilon_2, u_3)} \end{aligned}$$

This formulation $(\hat{P}_{\varepsilon_1, \varepsilon_2}^1)$ has the disadvantage that the efficient set is assumed to be known in advance, which contradicts the fact that a representation of Y_N should be computed.

However, the condition $x \in X_E$ can also be expressed implicitly with the help of a bilevel optimization problem, where the lower-level problem checks whether the current point is nondominated. A similar idea for checking efficiency of a proper face for linear problems was used in (Saym, 1996). Here, this idea is incorporated for general problems in the lower-level of the scalarization problem

$$(\hat{P}_{\varepsilon_1, \varepsilon_2}^{1, \text{bilevel}}) \quad \min_x z^x \tag{4a}$$

$$\text{s. t. } x \in X \tag{4b}$$

$$f(x) \in B(l, u)^{(\varepsilon_1, \varepsilon_2, u_3)} \tag{4c}$$

$$z^x := \max_{\tilde{x}} \sum_{i=1}^3 f_i(x) - f_i(\tilde{x}) \tag{4d}$$

$$\text{s. t. } \tilde{x} \in X \tag{4e}$$

$$f(\tilde{x}) \leq f(x) \tag{4f}$$

which defines a so called *min-max problem*, a special case of a bilevel optimization problem where the lower-level and upper-level objective functions coincide.

If subproblem $(\hat{P}_{\varepsilon_1, \varepsilon_2}^{1, \text{bilevel}})$ is feasible, the objective value is, due to constraint (4f), always non-negative. If the optimal objective value (assuming existence) attains the value 0, the corresponding $x \in X$ defines a nondominated point in $B(l, u)^{(\varepsilon_1, \varepsilon_2, u_3)}$, otherwise it can be implied that $Y_N \cap B(l, u)^{(\varepsilon_1, \varepsilon_2, u_3)} = \emptyset$.

Due to its bilevel structure, this problem is hard to solve and since such problems should be solved repeatedly as subproblems in the Box-Algorithm, we decided to investigate the simpler subproblem $(P_{\varepsilon_1, \varepsilon_2}^1)$, which can be solved using standard black-box solver (like CPLEX). Yet, this bilevel idea is a promising approach and will be pursued in the future.

References

- Barichard, V. and Hao, J.-K. (2003). A Population and Interval Constraint Propagation Algorithm. In Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., and Deb, K., editors, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 88–101. Springer Berlin / Heidelberg.
- Boland, N., Charkhgard, H., and Savelsbergh, M. (preprint, 2014a). A Simple and Efficient Algorithm For Solving Three Objective Integer Programs. *Optimization Online*.
- Boland, N., Charkhgard, H., and Savelsbergh, M. (preprint, 2014b). The L-Shape Search Method for Triobjective Integer Programming. *Optimization Online*.
- Chankong, V. and Haimes, Y. (1983). *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York.
- Dächert, K. and Klamroth, K. (2013). A linear bound on the number of tricriteria optimization problems. Technical report, Bergische Universität Wuppertal.
- Dhaenens, C., Lemesre, J., and Talbi, E. (2010). K-PPM: A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*, 200(1):45–53.
- Ehrgott, M. (2005). *Multicriteria optimization*. Springer, Berlin [u.a.], 2. edition.
- Ehrgott, M. and Tenfelde-Podehl, D. (2003). Computation of ideal and Nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151(1):119–139.
- Hamacher, H. W., Pedersen, C. R., and Ruzika, S. (2007). Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35(3):336–344.
- Kirlik, G. and Sayın, S. (2014). A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3):479–488.
- Laumanns, M., Thiele, L., and Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942.
- Ruzika, S. (2007). *On Multiple Objective Combinatorial Optimization*. PhD thesis, Technische Universität Kaiserslautern.
- Ruzika, S. and Wiecek, M. (2005). Approximation Methods in Multiobjective Programming. *Journal of Optimization Theory and Applications*, 126(3):473–501.
- Sayın, S. (1996). An algorithm based on facial decomposition for finding the efficient set in multiple objective linear programming. *Operations Research Letters*, 19(2):87–94.
- Sayın, S. (2000). Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560.
- Steuer, R. E. and Harris, F. W. (1980). Intra-set point generation and filtering in decision and criterion space. *Computers & Operations Research*, 7(1–2):41–53.