Finding Representative Systems for Discrete Bicriteria Optimization Problems by Box Algorithms

Horst W. Hamacher*, Christian Roed Pedersen†, Stefan Ruzika*

5th August 2005

Abstract

Given a discrete bicriteria optimization problem (DBOP), we propose two versions of an approximation procedure, the *box algorithm*, which results in a representation of the complete set of nondominated solutions by a finite representative system *Rep* satisfying the following quality features.

Its cardinality N and the accuracy Δ satisfy the relation $\mathcal{O}(\frac{a(R(z^1,z^2))}{\Delta})$, where $a(R(z^1, z^2))$ is the area of a starting box. Each of the points in *Rep* is a nondominated solution.

Rep can thus be used by Operations Research applicants instead of the complete set of nondominated solutions.

The paper concludes with a list of current research questions in the area of representative systems.

*Fachbereich Mathematik, Universität Kaiserslautern, Kaiserslautern, Germany; E-mail: {hamacher, ruzika}@mathematik.uni-kl.de. The research has been partially supported by Deutsche Forschungsgemeinschaft (DFG) grant HA 1737/7 "Algorithmik großer und komplexer Netzwerke" and by New Zealand's Julius von Haast award.

†Department of Operations Research, University of Aarhus, Aarhus, Denmark; E-Mail: roed@imf.au.dk.

Keywords: Multiple objective optimization; efficient solution; approximation; representative systems; box algorithm; quality measures.

1 Introduction

The *discrete* bicriteria optimization problem (DBOP) can be stated as

$$\begin{aligned} &\min \; f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} && (DBOP) \\ &\text{s.t.} \;\; x \in \mathcal{X} \end{aligned}$$

where $\mathcal{X}$ is a given discrete feasible set and $f : \mathcal{X} \longrightarrow \mathbb{Z}^2$ is a vector-valued objective or criterion function. $\mathcal{Y} := f(\mathcal{X})$ denotes the set of attainable outcomes or criterion vectors.

Minimizing a vector-valued objective function needs some more explanation since there is no canonical ordering defined in $\mathbb{Z}^2$. Throughout this article we use the Pareto concept of optimality for DBOPs which is based on the componentwise ordering of $\mathbb{Z}^2$ defined for any pair $y^1, y^2 \in \mathbb{Z}^2$ by

$$\begin{aligned} y^1 \leqq y^2 \quad &:\Leftrightarrow \quad y^1_k \leqq y^2_k \quad k = 1,2 \\ y^1 \leq y^2 \quad &:\Leftrightarrow \quad y^1_k \leqq y^2_k \quad k = 1,2 \quad \textit{and} \quad y^1 \neq y^2 \\ y^1 < y^2 \quad &:\Leftrightarrow \quad y^1_k < y^2_k \quad k = 1,2. \end{aligned}$$

A point $y^2 \in \mathbb{Z}^2$ is called *dominated* by $y^1 \in \mathbb{Z}^2$ if $y^1 \leq y^2$. According to the Pareto concept of optimality, the *efficient* or *Pareto set*, $\mathcal{X}_E$, and the *weakly efficient* or *weakly Pareto set*, $\mathcal{X}_{wE}$, are defined as

$$\begin{aligned} \mathcal{X}_E \quad &:= \quad \{x \in \mathcal{X} : \text{there exists no } \bar{x} \in \mathcal{X} : f(\bar{x}) \leq f(x)\} \\ \mathcal{X}_{wE} \quad &:= \quad \{x \in \mathcal{X} : \text{there exists no } \bar{x} \in \mathcal{X} : f(\bar{x}) < f(x)\}. \end{aligned}$$

The images

$$\mathcal{Y}_N := f(\mathcal{X}_E) \quad \text{and} \quad \mathcal{Y}_{wN} := f(\mathcal{X}_{wE})$$

of these sets under the vector-valued mapping f are called the *nondominated set* and the *weakly nondominated set*, respectively.

Since the problem of finding $\mathcal{Y}_N$ is, in general, intractable for discrete as well as continuous bicriteria optimization problems, the idea of computing a representation of $\mathcal{Y}_N$ arose. The term "representation" is not consistently used in the literature. For us, a representation *Rep* is a set of points in $\mathcal{Y}$ that is computed instead of the nondominated set $\mathcal{Y}_N$. Since, in general $Rep \neq \mathcal{Y}_N$, the question about the "quality" of such a representation naturally occurs. Although quality measures of a representation (sometimes also called approximation) of the nondominated set were implicitly used in some algorithms (for example as stopping criteria), the article by Sayin ([13]) seems to be the first which explicitly deals with the question of how to measure the quality of a representation. For the case of discrete problems, on which we focus in this paper, we evaluate our algorithms based on the following quality attributes of our representative system.

1. Cardinality
2. Accuracy
3. Error
4. Cluster Density

Here, *cardinality* refers to the fact that our representative system should be "reasonably" small, since the computation of every representing point requires a certain effort, usually related to the solution of a (single criterion) optimization problem. An *accurate* representation should mirror the complete set of nondominated solutions satisfying some quality measure. Informally, this excludes a situation, where large parts of the set of nondominated solutions are not represented by any point in the representative system. With *error* we measure the distance between the representing points and nondominated points. If all representing points are contained in $\mathcal{Y}_N$, then there is no error. The *cluster density* is closely related to the cardinality issue. Obviously, it should be avoided that *Rep* contains large clusters of points which represent the same subset of nondominated solutions.

The reader should be aware that these quality issues are conflicting by nature. For example, the smaller the cardinality of a representation, the larger the "gaps", i.e., the more inaccurate the representation.

For continuous bicriteria optimization problems, the *sandwich algorithm* (see Burkard et al. [1] and [2] and Fruhwirth et al. [6]) and the *gauge algorithm* (see Klamroth et al. [9]) are among the most sophisticated methods for generating representations of $\mathcal{Y}_N$. Both algorithms iteratively refine a piecewise linear structure by generating new approximating points until this structure is close enough to $\mathcal{Y}_N$. For a more complete overview of methods for generating representations and a discussion on which of the above quality attributes are addressed by the sandwich and the gauge algorithm, the reader is referred to the survey paper of Ruzika and Wiecek [12].

For discrete bicriteria optimization problems, the literature on exact approaches for computing representations is scarce. In fact, to the best of our knowledge, the gauge algorithm is the only exact algorithm. Although this algorithm is originally intended for continuous problems, it is also applicable to discrete problems. On the other hand, there exists a large variety of heuristic methods (see, for instance, the overview in Ehrgott and Gandibleux [4]). Note that these heuristic approaches often do not address the quality attributes mentioned above, in particular in regard to the error of the representation.

Based on the preceding list of quality measures, we develop in this paper a generic method for computing representations of discrete bicriteria optimization problems, the *box method*. The representation consists of a collection of nondominated points, i.e. $Rep \subseteq \mathcal{Y}_N$. By design, our algorithms will therefore satisfy the error quality criterion with a zero error.

Each point is associated with a rectangle (or *box*) and represents all nondominated points within this box. The geometrical features of the rectangles can thus be used to measure the preceding quality attributes.

At any stage of the algorithm, the collection of boxes contains all the nondominated set. Hence, the rectangles bound the nondominated set, and thus also show regions containing no nondominated points. Representing points are generated with a modification of the ε-constraint method. This method is capable of generating any nondominated point with appropriate choices of parameters. Furthermore, optimal solutions of the modified ε-constraint problem are known to be nondominated.

Rectangles have been used previously in the context of bicriteria optimization problems. An interactive method by Payne and Polak [11], constructs a nested family of rectangles based on a decision makers choice of most preferred rectangle during the algorithm. Payne [10] develops a chain of rectangles containing all nondominated points, ensuring a certain quality aspect to be fulfilled at termination of the algorithm. Both papers solve iteratively nonlinear programs. Instead of focussing on the criterion space, one can also work in the decision space. The *Big Square – Small Square* (BSSS) method by Hansen et al. [8] has been applied among others by Skriver and Andersen [14] to eliminate rectangles containing only non-Pareto locations for a bicriterion semi-obnoxious location problem.

The remainder of the paper is organized as follows. In Section 2 we introduce a variant of the well-known ε-constraint method which will be employed for generating representing points. Two versions of the box algorithm for computing a discrete representation with provable quality features are proposed in Section 3. The algorithms are discussed and their quality properties are linked to existing literature in Section 4. We conclude in Section 5 the paper with comments on ongoing and future research in this area.

2 The lexicographic ε-constraint method

The scalarization that we will use intensively in the remainder of this paper is a lexicographic variant of the well-known ε-constraint scalarization (see, e.g., Chankong and Haimes [3]). We refer to the following mathematical program as the *lexicographical ε-constraint problem.*

$$\begin{aligned} \text{lex min} \quad & \begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix} \\ \text{s.t.} \quad & f_1(x) \le \varepsilon \\ & x \in \mathcal{X}. \end{aligned} \tag{P_ε}$$

In P_ε the vector $(f_2(x), f_1(x))^T$ is lexicographically minimized, and its feasible set is the feasible set of the bicriteria optimization problem $\mathcal{X}$ with an additional constraint bounding the f_1-values from above. Since the lexicographical ordering is complete (i.e., any two points are comparable) a lexicographical objective function is as easy to optimize as a single criterion objective.

The lexicographical ε-constraint problem has, however, several desirable properties which are very useful in constructing a representation of the nondominated set. In particular, the next result shows that the lexicographic objective function yields (strongly) efficient solutions - opposed to weakly efficient ones in the classical ε-constraint approach.

Theorem 2.1 *Let $\hat{x} \in \mathcal{X}$ be an optimal solution of P_ε. Then $\hat{x} \in \mathcal{X}_E$.*

Proof:
Let $x' \in \mathcal{X}$ with $f(x') \leqq f(\hat{x})$. In particular, $f_1(x') \leqq f_1(\hat{x}) \leqq \varepsilon$, such that x' is feasible for P_ε. The definition of the lexicographical ordering and the optimality of $\hat{x}$ for P_ε excludes that $f(x') \leq f(\hat{x})$, hence $\hat{x} \in \mathcal{X}_E$. $\square$

Conversely, any efficient solution can be obtained with the lexicographic ε-constraint scalarization with an adequate choice of ε.

Theorem 2.2 *Let $\hat{x} \in \mathcal{X}_E$. Then $\varepsilon := f_1(\hat{x})$, yields $\hat{x}$ as an optimal solution for P_ε.*

Proof:
The proof follows directly from the same result of the classical ε-constraint scalarization, see for instance Chankong and Haimes [3]. $\square$

In particular, the one-to-one correspondence between solutions of P_ε and $\mathcal{X}_E$ implies that we get supported as well as unsupported Pareto solutions. This is not achievable by the weighted sum method which is by far the most utilized scalarization in the literature.

An additional advantage of using the lexicographic ε-constraint method is the fact that we maintain the original objective functions. This is an appealing feature, especially if the original objective functions possess structural properties like linearity or convexity. To the best of our knowledge, all scalarization techniques (for a list of the most common scalarizations we refer to Chapter 17 in [5]) which preserve linearity of the objective functions and which are capable of generating all nondominated points for discrete problems, modify the constraint set by at least one constraint. Therefore, the additional constraint induced by the lexicographic ε-constraint method can be viewed as smallest possible modification.

3 Two versions of the box algorithm

In this section we propose two versions of the box algorithm for computing a representation (in $\mathcal{Y}$) of the nondominated set $\mathcal{Y}_N$ of a discrete bicriteria optimization problem. Both versions allow controlling the quality attributes of a representation as proposed in the introduction.

3.1 Initialization and update

Before explaining the details, we outline the major ideas of the box algorithm. Initially, we compute the two lexicographical optimal solutions. These points determine a rectangle (the *starting box*) containing the complete nondominated set. In the following we discard rectangular pieces of the starting box based on additional information obtained by iteratively solving P_ε with adequately chosen values for ε. This yields a collection of rectangles or *boxes* containing the nondominated set in each stage of the algorithm. Furthermore, we know for each box a nondominated point representing the set of nondominated points inside this box. The box algorithm stops based on a predetermined accuracy which is measured by the biggest of the remaining boxes.

More detailed, the box algorithm works as follows. Initially, we determine both lexicographic minima of the bicriteria optimization problem

$$z^1 := \begin{pmatrix} z_1^1 \\ z_2^1 \end{pmatrix} := \text{lex} \min_{x \in \mathcal{X}} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} \text{ and } z^2 := \begin{pmatrix} z_1^2 \\ z_2^2 \end{pmatrix} := \text{lex} \min_{x \in \mathcal{X}} \begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix}.$$

Obviously, $\mathcal{Y}_N$ is a subset of $R := R(z^1, z^2)$, the rectangle with upper left and lower right vertex z^1 and z^2, respectively. This is the *starting box* of our algorithms. In general, we will use in the following the notation $R(y^1, y^2)$ to denote the rectangle having $y^1 = \begin{pmatrix} y_1^1 \\ y_2^1 \end{pmatrix} \in \mathbb{Z}^2$ as upper left and $y^2 = \begin{pmatrix} y_1^2 \\ y_2^2 \end{pmatrix} \in \mathbb{Z}^2$ as lower right vertex. Moreover, let $a(R(y^1, y^2)) := (y_1^2 - y_1^1) \cdot (y_2^1 - y_2^2)$ denote the area of the rectangle $R(y^1, y^2)$.

Initially, the starting box $R(z^1, z^2)$ contains the complete nondominated set $\mathcal{Y}_N$. Whenever additional nondominated points become known during the execution of the algorithm, one of the boxes will be split up into several smaller rectangles -

while maintaing the inclusion property of $\mathcal{Y}_N$. We will stop when the area of the largest of these boxes is smaller than some given accuracy $\Delta > 0$. We refer to such a collection of boxes as a *Δ-representation of $\mathcal{Y}_N$*. The lower right corner point of each of the rectangles is a *representing point* and will be added to the representing system.

In the following, we consider a box with $a(R(y^1, y^2)) > \Delta$, such that the representation has to be locally updated in $R(y^1, y^2)$. Consider P_ε with $\varepsilon := \lfloor \frac{y_1^1 + y_1^2}{2} \rfloor$. Let $x^* \in \mathcal{X}$ be optimal for P_ε and let $z^* := (f_1(x^*), f_2(x^*)) = f(x^*)$. Due to Theorem 2.1, z^* is nondominated. Using the point z^* and ε, we divide $R(y^1, y^2)$ into five rectangles

$$R_1 := R(y^1, z^*) \quad R_2 := R(p^1, p^4) \quad R_3 := R(p^2, p^6) \quad R_4 := R(p^3, p^7) \quad R_5 := R(p^5, y^2)$$

(see Figure 1), where

$$p^1 := \begin{pmatrix} z_1^* \\ y_2^1 \end{pmatrix} \quad p^2 := \begin{pmatrix} \varepsilon + 1 \\ y_2^1 \end{pmatrix} \quad p^3 := \begin{pmatrix} y_1^1 \\ z_2^* \end{pmatrix} \quad p^4 := \begin{pmatrix} \varepsilon \\ z_2^* \end{pmatrix}$$

$$p^5 := \begin{pmatrix} \varepsilon + 1 \\ z_2^* - 1 \end{pmatrix} \quad p^6 := \begin{pmatrix} y_1^2 \\ z_2^* \end{pmatrix} \quad p^7 := \begin{pmatrix} \varepsilon \\ y_2^2 \end{pmatrix}.$$

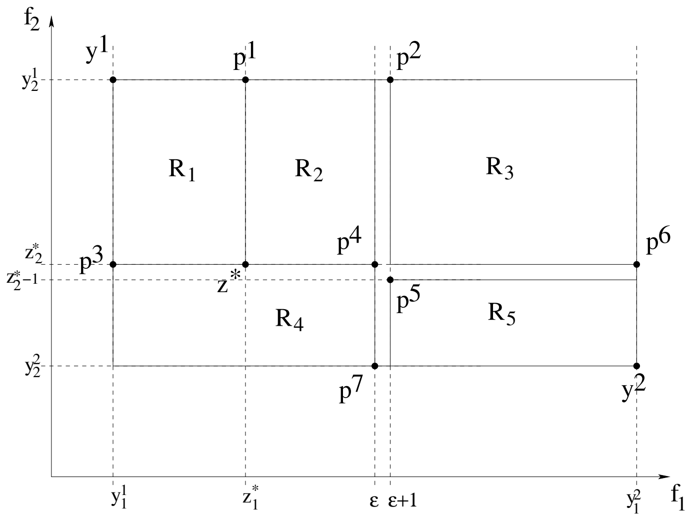

Figure 1: Updating of a rectangle $R(y^1, y^2)$.

Notice that the points p^2 and p^5 have first coordinate one unit to the right of the ε-constraint. Furthermore the point p^5 has second coordinate one smaller than the second objective value of z^*. This is due to the fact, that no feasible nondominated point can have non-integral coordinates. In general, $p^i \notin \mathcal{Y}$, $i = 1, \ldots, 7$, i.e., the points p^i are not necessarily feasible criterion vectors. The rectangles R_1, R_2, and R_4 can degenerate to a line, in which case they are considered to have zero area.

Proposition 3.1 *R_2, R_3, and R_4 can be eliminated, since*

a) $(R_2 \cup R_3) \cap \mathcal{Y}_N \subseteq \{y^1, z^*\}$, *and*

b) $R_4 \cap \mathcal{Y}_N \subseteq \{z^*\}$.

Proof:
a) z^* is dominating any feasible point in $R_2 \cup R_3$.
b) Any feasible point in R_4 contradicts the optimality of z^* for P_ε. □

Due to the previous result, there do not exist nondominated points in R_2, R_3, R_4, other than z^* (and possibly y^1). Consequently, we can formulate the following corollary.

Corollary 3.1

$$\mathcal{Y}_N \cap R(y^1, y^2) \subseteq R_1 \cup R_5$$

It is easy to see, that the following result comparing the size of rectangles R_1 and R_5 to the size of rectangle $R(y^1, y^2)$ holds true.

Proposition 3.2

$$a(R_1) + a(R_5) \leq \frac{1}{2} a(R(y^1, y^2))$$

After having obtained z^* as a new nondominated point, the representation is updated associating z^* with R_1 and y^2 with R_5. By Proposition 3.2 we see that computing z^* has locally improved the representation by a factor of 2.

Corollary 3.2 *After computing z^* and adding it to the representation, the representation's accuracy improved in $R(y^1, y^2)$ by a factor of 2.*

3.2 The a posteriori algorithm

In this subsection we iteratively apply the updating procedure from Section 3.1 to place ε-constraints in a rectangle having area bigger than Δ. We assume that Δ is given as input in our algorithm. Since it is the measure of the size of our boxes, we may also leave its determination to the decision maker or compute it after Step 4 as a given percentage of the starting box.

Since the value of the ε-constraint is chosen *after* the given rectangle is identified, this method is called *a posteriori* algorithm. A pseudo code description is given in Algorithm 1.

Algorithm 1 A posteriori algorithm.

Input: A discrete bicriteria optimization problem, $\Delta > 0$

Output: A representation $Rep \subseteq \mathcal{Y}_N$ with accuracy Δ.

1: $S \leftarrow \emptyset$.
2: $Rep \leftarrow \emptyset$.
3: Compute the lexicographic minima z^1 and z^2 and $a(R(z^1, z^2))$.
4: $Rep \leftarrow \{z^1, z^2\}$.
5: $S \leftarrow \{R(z^1, z^2))\}$.
6: **while** $S \neq \emptyset$ **do**
7:     Choose the largest rectangle $R(y^1, y^2) \in S$.
8:     $S \leftarrow S \setminus \{R(y^1, y^2)\}$.
9:     Solve P_ε with $\varepsilon := \left\lfloor y_1^1 + \frac{y_1^2 - y_1^1}{2} \right\rfloor$ and obtain optimal solution $z^* \in \mathcal{Y}_N$.
10:     $Rep \leftarrow Rep \cup \{z^*\}$.
11:     **if** $a(R(y^1, z^*)) > \Delta$ **then**
12:         $S \leftarrow S \cup \{R(y^1, z^*)\}$.
13:     **end if**
14:     **if** $a(R(p^5, y^2)) > \Delta$ **then**
15:         $S \leftarrow S \cup \{R(p^5, y^2)\}$.
16:     **end if**
17: **end while**

Note that the a posteriori algorithm is flexible in the sense that it always chooses the largest rectangle and the representation is hence only refined where it needed.

The following theorem establishes finiteness and the complexity of the a posteriori algorithm.

Theorem 3.1 *In finitely many steps, the a posteriori algorithm yields a Δ-representation of $\mathcal{Y}_N$ in which all representing points are nondominated. More specifically, the algorithm performs at most $\mathcal{O}(\frac{A}{\Delta})$ many iterations, where A is the volume of the starting box $R(z^1, z^2)$.*

Proof:

As long as the representation is not a Δ-representation, the algorithm processes in each iteration the largest rectangle that does not meet the accuracy of Δ. Due to Corollary 3.2, the resulting rectangle(s) have after the update an area which is at most half the size as the original area. The algorithm is therefore finite. Due to Theorem 2.1, all representing points are nondominated.

More specifically, the algorithm produces after $2^k - 1$ iterations no more than 2^k rectangles, each of which have an area of at most $\frac{A}{2^k}$. Since $k^* = \lceil \log_2(\frac{A}{\Delta}) \rceil$ is the smallest integer satisfying $\frac{A}{2^{k^*}} \leq \Delta$, an upper bound on the number of iterations to obtain a Δ-representation is $2^{k^*} - 1 = 2^{\lceil \log_2(\frac{A}{\Delta}) \rceil} - 1$, which is $\mathcal{O}(\frac{A}{\Delta})$. □

A direct consequence of Theorem 3.1 is the following result.

Corollary 3.3 *The a posteriori algorithm runs in $\mathcal{O}(T_1 + \frac{a(R(y^1, y^2))}{\Delta} T_2)$ where T_1 and T_2 is the time needed to compute a lexicographic minimum and to evaluate P_ε, respectively.*

Observe that T_1 and T_2 depend on the specific problem under consideration.

3.3 The a priori algorithm

Rather than deciding on the value of ε for the next ε-constraint problem *after* each iteration, we compute in the next algorithm *a priori* a number of equidistant values for ε based on the value of Δ. The solution of the corresponding ε-constraint problems yields a Δ-accurate representation for any given Δ.

Consider

$$\varepsilon_j := \left\lfloor z_1^1 + \frac{j}{k+1}(z_1^2 - z_1^1) \right\rfloor$$

for $j = 1, \ldots, k$. Let $\hat{z}^j$ denote the image of the optimal solution to P_{ε_j}. Note that the points $\hat{z}^j$, $j = 1, \ldots, k$, are not necessarily distinct. If $\hat{z}^{j+1} = \hat{z}^j$, we define $p^{j+1} := \hat{z}^j$, $j = 1, \ldots, k-1$. Else, let $p^j := (\varepsilon_j + 1, \hat{z}_2^j - 1)$, $j = 1, \ldots, k$. The points $\{p^j\}$ are used to define rectangles, but they are not necessarily feasible in $\mathcal{Y}$. Figure 2 illustrates these definitions.

The same reasoning concerning dominance and feasibility as in Proposition 3.1 and Corollary 3.1 applies here as well. Therefore, we can conclude that

$$\mathcal{Y}_N \subseteq R(z^1, \hat{z}^1) \cup \bigcup_{j=1}^{k-1} R(p^j, \hat{z}^{j+1}) \cup R(p^k, z^2)$$

that means $\mathcal{Y}_N$ is contained in a collection of at most $k+1$ rectangles. For each rectangle, its lower right corner point is nondominated.

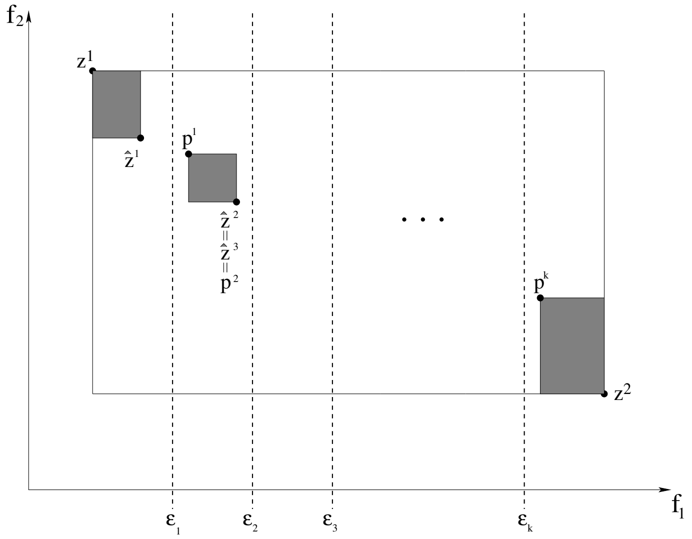

Figure 2: The representation after having solved P_{ε_1}, P_{ε_2} $\ldots$, P_{ε_k}.

The following proposition evaluates the reduction rate of the unknown area of interest.

Proposition 3.3 *Suppose the algorithm solves k ε-constraint problems with equidistant ε-constraints. Then the sum of the areas of the resulting rectangles is at most $\frac{1}{k+1}a(R(z^1, z^2))$.*

Proof:

$$\begin{aligned}
& a(R(z^1,\hat{z}^1)) + \sum_{j=1}^{k-1} a(R(p^j,\hat{z}^{j+1})) + a(R(p^k,z^2)) \\
= \quad & (z_2^1 - \hat{z}_2^1)(\hat{z}_1^1 - z_1^1) + \sum_{j=1}^{k-1}(p_2^j - \hat{z}_2^{j+1})(\hat{z}_1^{j+1} - p_1^j) + (p_2^k - z_2^2)(z_1^2 - p_1^k) \\
\leq \quad & \frac{1}{k+1}(z_1^2 - z_1^1)\Big(z_2^1 - \hat{z}_2^1 + \sum_{j=1}^{k-1}(\hat{z}_2^j - \hat{z}_2^{j+1}) + \hat{z}_2^k - z_2^2\Big) \\
= \quad & \frac{1}{k+1}a(R(z^2,z^1))
\end{aligned}$$

□

It should be emphasized that the *total* area of the rectangles is at most $\frac{1}{k+1}a(R(z^1, z^2))$. Hence the biggest rectangle in the representation has area of at most $\frac{1}{k+1}a(R(z^1, z^2))$. We thus get the next result.

Theorem 3.2 *To achieve an accuracy of Δ in the a priori algorithm $k \leq \lceil \frac{a(R(z^1,z^2))}{\Delta} \rceil - 1$ solutions of P_ε have to be computed.*

The algorithm is explicitly stated in Algorithm 2 and illustrated in Figure 2.

4 Quality of the box representation

In this section we will address for both algorithms the quality issues introduced in Section 1.

Cardinality Due to Theorems 3.1 and 3.2, the number of representing points is at most $2^{\lceil \log_2(\frac{a(R(z^1,z^2))}{\Delta}) \rceil} + 1$ and $\lceil \frac{a(R(z^1,z^2)}{\Delta} \rceil + 1$ for the a posteriori algorithm and the a priori algorithm, respectively. However, in both cases, a worst-case analysis is used to obtain

Algorithm 2 A priori algorithm.

Input: A discrete bicriteria optimization problem, $\Delta > 0$

Output: A representation $Rep \subseteq \mathcal{Y}_N$ with accuracy Δ.

1: $Rep \leftarrow \emptyset$.
2: Compute the lexicographic minima z^1 and z^2 and $a(R(z^1, z^2))$.
3: $Rep \leftarrow \{z^1, z^2\}$.
4: $k \leftarrow \lceil \frac{a(R(z^1,z^2))}{\Delta} \rceil - 1$
5: **for** $j = 1 \dots k$ **do**
6: $\quad \varepsilon_j := \lfloor z_1^1 + \frac{j}{k+1}(z_1^2 - z_1^1) \rfloor$.
7: $\quad$ Solve P_{ε_j}.
8: $\quad Rep \leftarrow Rep \cup \{\hat{z}^j\}$.
9: **end for**

these bounds. It can be expected, that the number of generated points is much smaller in practice since solutions can be optimal for several ε-constraint problems simultaneously.

Alternatively, the two algorithms can also be formulated in such a way, that the cardinality N of the representative system is part of input data instead of Δ. Then the accuracy Δ is a function of N, i.e., the representation is Δ-accurate with $\Delta = \frac{a(R(z^1,z^2)}{N-1}$ and $\Delta = \frac{a(R(z^1,z^2)}{N-2}$ for the two algorithms, respectively.

Accuracy Both algorithms have as termination criterion a measure for the accuracy and therefore, this issue has already been discussed in Section 3.2 and Section 3.3.

Using the different accuracy measure of [13] it is easy to show that Rep is also a d_Δ-representation in the sense of [13] when using l_∞ to measure distances.

A further definition of accuracy from the literature measures the percentage ϵ of distance between any nondominated point from the nearest representing point in both coordinates. Using this definition, an approximation Rep is called ϵ-approximation if $\mathcal{Y}_N$ is dominated by $(1-\epsilon)Rep$ (see e.g. [16]). The rectangle representation attained by any of the two algorithms in this paper, yields an ϵ of $\max_{y \in Rep, y \in R(z,y)}\{\frac{y_1 - z_1}{y_1}\}$, where y is a representing point in the rectangle $R(z, y)$.

Finally, it is obvious that both algorithms are exact if Δ is sufficiently small, e.g. $\Delta = 1$.

Error Extending ideas from [13], we can define the error formally as

$$error = \max_{z \in Rep} \min_{y \in \mathcal{Y}_N} ||z - y||_p$$

where $||.||_p$ denotes the l_p norm. In both algorithms, only nondominated points are generated and therefore, the error is always zero for any p. Note that $\min\{accuracy, error\}$ is the Hausdorff distance which is used in a variety of algorithms for computing a representation of $\mathcal{Y}_N$ (for continuous problems) ([6, 17])

Cluster Density Depending on the application, one can apply filtering procedures (see e.g. [15]) at the termination of either of the two algorithms to reduce the number of representing points in any rectangle of size Δ to a desirable amount of points. This filtering obviously reduces the cardinality of the representation while maintaining the Δ-accuracy.

5 Conclusion and Further Research

In this paper we have shown, how an iterative update of boxes containing the set of nondominated solutions leads to a representative system for discrete bicriteria optimization problems. We have shown that in both versions of the box algorithm - a priori and a posteriori - the cardinality N of the resulting repesentative set and the accuracy Δ satisfy the relation $\mathcal{O}(\frac{a(R(z^1,z^2))}{\Delta})$ such that we can either input the accuray and compute the needed cardinality or vice versa in our box algorithm.

Several variations and extentions of the box approach are possible.

The first addresses the way, accuracy is measured. Instead of using the area $a(R(y_1, y_2))$ of the boxes, other means, such as the maximal diagonal or the maximal side among the boxes, might prove useful in certain applications. This can be achieved with the same general idea as presented in Section 3. If the maximal length is attained in the second component the ε-constraint is introduced for function f_2 instead of f_1. The corresponding analysis of the number of iterations is formally more complicated, but rather straightforward.

Although the algorithms are presented in the context of DBOPs, minor changes also allow their application to nonlinear bicriteria optimization problems. Numerical tests are currently under research to show, whether this approach is competitive with the sandwich approach of Fruhwirth et al. [6].

Another appealing extension is the generalization of the algorithms to discrete problems with more than two objective functions. If we assume that, without loss of generality, $Q = 2^k$ many objective functions are given we can iteratively apply the bicriteria approach to pairs of objectives and get in this way a representative system. The analysis of the resulting procedure regarding the quality measures of Section 1 is under research. Another approach is to use Q-dimensional boxes instead of 2-dimensional in the bicriteria case. While this approach is conceptually very similar to the bicriteria situation, the computation of the starting box requires the solution of $Q!$ many lexicographical problems. This is feasible for small Q, but no longer so for large Q - a situation which very often occurs in real-world problems.

The complexity of both versions of the box algorithm depends on the complexity of solving the ε-constraint problems which occur as subproblems. Interesting research questions concern the usage of the solution of one of these problems in solving another one with modified ε, and ways on how to measure the effect of an approximation of the ε-constraint problems inside the box algorithm. Current research in this area concentrates on bicriteria flow problems for which currently no specialized algorithm to compute a representative system exists (see Hamacher et al. [7]).

References

[1] R.E. Burkard, H.W. Hamacher, and G. Rote. Sandwich approximation of univariate convex functions with an application to separable convex programming. *Naval Research Logistics Quarterly*, 38(6):911–924, 1991.

[2] R.E. Burkard, G. Rote, G. Ruhe, and N. Sieber. Algorithmische Untersuchungen zu

bikriteriellen kostenminimalen Flüssen in Netzwerken. *Wissenschaftliche Zeitschrift der technischen Hochschule Leipzig*, 13(6):333–341, 1989.

[3] V. Chankong and Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science, New York, 1983.

[4] M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Kluwer Academic Publishers, Boston, Massachusetts, 2002.

[5] J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis - State of the Art Surveys*. Springer, New York, 2005.

[6] B. Fruhwirth, R.E. Burkard, and G. Rote. Approximation of convex curves with application to the bicriterial minimum cost flow problem. *European Journal of Operational Research*, 42:326–338, 1989.

[7] H.W. Hamacher, C.R. Pedersen, and S. Ruzika. Multiple objective minimum cost flow problems: A review. Working Paper 2005/1, 2005. Department of Operations Research, University of Aarhus, Denmark.

[8] P. Hansen. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple criteria decision making - Theory and Applications*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 109–127. Springer Verlag, Heidelberg, 1980.

[9] K. Klamroth, Jorgen Tind, and Margaret M. Wiecek. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56(3):413–437, 2002.

[10] A.N. Payne. Efficient approximate representation of bi-objective tradeoff sets. *Journal of the Franklin Institute*, 330(6):1219–1233, 1993.

[11] A.N. Payne and E. Polak. An interactive rectangle elimination method for biobjective decision making. *IEEE Transactions on Automatic Control*, 25(3):421–432, 1980.

[12] Stefan Ruzika and Margaret Wiecek. A survey of approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3), 2005.

[13] Serpil Sayin. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87:543–560, 2000.

[14] A.J.V. Skriver and K.A. Andersen. The bicriterion semi-obnoxious location (BSL) problem solved by an ε-approximation. *European Journal of Operational Research*, 146:517–528, 2003.

[15] R.E. Steuer and F.W Harris. Intra-set point generation and filtering in decision and criterion space. *Computers and Operations Research*, 7:41–53, 1980.

[16] A. Warburton. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1):70–79, 1987.

[17] X.Q. Yang and C.J. Goh. A method for convex curve approximation. *European Journal of Operational Research*, 97:205–212, 1997.