Efficient Computation of the Approximation Quality in Sandwiching Algorithms

Ina Lammel, Karl-Heinz Küfer, Philipp Süss

Department of Optimization
Fraunhofer Institute for Industrial Mathematics (ITWM)
Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
{ina.lammel, karl-heinz.kuefer, philipp.suess}@itwm.fraunhofer.de

October 4, 2023

Abstract

Computing the approximation quality is a crucial step in every iteration of Sandwiching algorithms (also called Benson-type algorithms) used for the approximation of convex Pareto fronts, sets or functions. Two quality indicators often used in these algorithms are polyhedral gauge and epsilon indicator. In this article, we develop an algorithm to compute the polyhedral gauge and epsilon indicator approximation quality more efficiently. We derive criteria that assess whether the distance between a vertex of the outer approximation and the inner approximation needs to be recalculated. We interpret these criteria geometrically and compare them to a criterion developed by Dörfler et al. for a different quality indicator using convex optimization theory. For the bi-criteria case, we show that only two linear programs need to be solved in each iteration. We show that for more than two objectives, no constant bound on the number of linear programs to be checked can be derived. Numerical examples illustrate that incorporating the developed criteria into the Sandwiching algorithm leads to a reduction in the approximation time of up to 94 % and that the approximation time increases more slowly with the number of iterations and the number of objective space dimensions.

1 Introduction

In multicriteria optimization problems, several objectives are to be optimized simultaneously. A common task is to compute the nondominated set, also

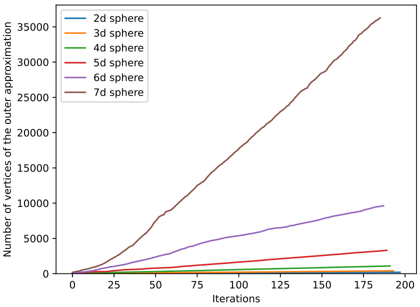


Figure 1: Number of vertices of the outer Sandwiching approximation in dependence of the number of iterations for spheres of different dimension.

called Pareto front. Since the nondominated set of convex multicriteria optimization problems generally has an uncountable number of nondominated solutions, approximation algorithms are usually used. One common technique is to approximate a convex bounded Pareto front using an inner and outer polyhedral approximation. Algorithms of this type are often called (Simplicial) Sandwiching algorithms (e.g.[4], [15], [21], [24]) or Benson-type algorithms (e.g. [1], [9], [11], [19]). In these algorithms, an inner and an outer polyhedral approximation are constructed using nondominated points and their gradients. By computing additional nondominated points, the inner and outer approximations are improved to describe the Pareto front more accurately.

The same algorithmic ideas are also applied to the approximation of convex functions (e.g. [22]) and convex sets (e.g. [13], [14]). Recently, the idea of Sandwiching algorithms has been applied to the approximation of multiple convex Pareto fronts, e.g. for multiobjective mixed-integer convex optimization in [5] (bi-objective problems) and [17] (general number of objectives).

A crucial step in each iteration of these algorithms is the decision where to place a new point. Often, some kind of measure is used that computes the distance between the inner and the outer approximation as an upper bound on the approximation quality of the inner and outer approximation respectively. A new nondominated point is then computed so that it decreases the distance between the inner and outer approximation where it is the largest.

However, the number of locations where this can occur increases quickly with the number of iterations, especially for high numbers of objective functions (see Figure 1). E.g., an outer approximation of a 7d sphere using 200 points has more than 36000 vertices. Therefore, computing the approximation quality is usually the most time-consuming element of the whole approximation algorithm if solving the scalarization problems is not too difficult.

Different strategies have been introduced in the literature to tackle or circumvent this problem. Some algorithms do not compute the approximation quality in every iteration but instead choose a vertex of the outer approximation randomly that then serves as a starting point for the computation of a new Pareto point (e.g. [1], [11], [15], [19]). Other algorithms compute an estimate of the approximation quality, e.g. [24] and [4].

In other algorithms, the approximation quality is computed exactly. While [6] and [21] compute the distance between the inner and outer approximation for every vertex of the outer approximation, [9] avoids performing the distance computation in many cases. This result is based on the observation that in many cases, the distance between a vertex of the outer approximation and the inner approximation does not change from one iteration to the other. In [9], a variant of the Sandwiching algorithm is studied that uses the Hausdorff distance (defined using the Euclidean norm) between the inner and outer approximation as the quality measure. A criterion is introduced that checks for a vertex of the outer approximation whether its Euclidean distance to the inner approximation is the same as in the last iteration.

Many measures which are commonly used to assess the quality of Sandwiching approximations are based on the Hausdorff metric with different metrics used to compute the distance between elements of the respective sets. Examples are the epsilon indicator, used e.g. in [4] and [21] and the polyhedral gauge, used e.g. in [15] and [24]. They have the advantage that the computation of their distance between a vertex of the outer and the inner approximation can be modelled as a linear optimization problem. Additionally, the epsilon indicator value can be interpreted directly on the objective functions.

In this article, we will derive criteria which determine for a vertex of the outer approximation whether the epsilon indicator or polyhedral gauge distance to the inner approximation has to be recalculated after a new nondominated point has been added to the Sandwiching approximation. We exploit the property that our quality indicators can be modelled as linear optimization problems and obtain the criteria using the concept of reduced costs from linear optimization theory. We interpret these criteria geometrically and compare them to the criterion developed in [9] for the Euclidean norm in the context of a different Sandwiching variant in Section 5. In Section 7, we will show that computing the epsilon indicator and polyhedral gauge approximation quality of a Sandwiching approximation of a bi-criteria optimization problem can be reduced to solving two small linear programs per Sandwiching iteration. We demonstrate that for problems with more than two criteria, in general no constant bound can be derived. We illustrate the resulting speed-up of the Sandwiching algorithm using some numerical examples in Section 8.

2 Preliminaries

2.1 Bounded convex multiobjective optimization

A convex multiobjective optimization problem is defined as

$$\begin{aligned} &\min f(x) = (f_1(x), \ldots, f_d(x)) \\ &\text{subject to } x \in \mathcal{X}, \end{aligned} \tag{2.1}$$

(Definition 2.1.3 of [20]) where $f(x)$ denotes the vector of d convex objective functions $f_i : \mathbb{R}^k \to \mathbb{R}$, and the decision vectors $x \in \mathbb{R}^k$ are elements of the convex feasible set $\mathcal{X}$. We assume that the problem is solved with respect to a pointed, convex, polyhedral ordering cone $\mathcal{C}$ fulfilling $0 \in \mathcal{C}$ and $\mathcal{C} \supset \mathbb{R}^d_{\geq}$.

A feasible solution $\hat{x} \in \mathcal{X}$ is called *efficient* if there is no other $x \in \mathcal{X}$ such that $f(x) \leq f(\hat{x})$. If $\hat{x}$ is efficient, $f(\hat{x})$ is called *nondominated* [10]. As in [20], we also denote nondominated points as Pareto optimal and the nondominated set as Pareto front. A feasible solution $\hat{x} \in \mathcal{X}$ is called *weakly efficient* if there is no $x \in \mathcal{X}$ such that $f(x) < f(\hat{x})$, i.e. $f_i(x) < f_i(\hat{x})\ \forall i = 1, \ldots, d$. If $\hat{x}$ is weakly efficient, $f(\hat{x})$ is called *weakly nondominated* [10].

The weighted sum scalarization of problem is given by ((3.3) of [10])

$$\min_{x \in \mathcal{X}} \sum_{i=1}^{d} \lambda_i f_i(x) \tag{2.2}$$

with values $\lambda = (\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^d_{\geq}$. Well-known results include that for a convex optimization problem, every solution of the weighted sum problem is weakly nondominated and that every element of the nondominated set can be computed by solving weighted sum scalarization problems for $\lambda \in \mathcal{C}^* \backslash \{0\}$ where $\mathcal{C}^*$ denotes the dual cone (e.g. [10]).

A vector $k \in \mathbb{R}^d \backslash \{0\}$ is called a direction of the cone $\mathcal{C}$ if $\{c + \alpha k \in \mathbb{R}^d : c \in \mathcal{C}, \alpha > 0\} \subset \mathcal{C}$. The set of extreme directions of a convex cone is a set of directions such that all directions of the cone lie in their convex hull. If the solution of the weighted sum scalarization problem (2.2) for all extreme directions of the dual cone $\mathcal{C}^*$ of $\mathcal{C}$ is bounded, the solution of the convex multiobjective optimization problem is bounded.

2.2 Some concepts and results from linear optimization

Let a *linear optimization problem (LP)* be defined as

$$\min\ c^T x \text{ s.t. } Ax = b,\ x \geq 0 \tag{2.3}$$

with $c \in \mathbb{R}^n,\ A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m,\ m \leq n$ and A fulfils $\operatorname{rank}(A) = m$.

Definition 2.1 (p. 17 of [12]). *Consider the linear program* (2.3) *with basic feasible solution $x = (x_B, x_N)$. We use the following notation: $x_{B(1)}, \ldots, x_{B(m)}$ denotes the basic variables and $x_{N(1)}, \ldots, x_{N(n-m)}$ the non-basic variables, c is partitioned into $c_B = (c_{B(1)}, \ldots, c_{B(m)})$ and $c_N = (c_{N(1)}, \ldots, c_{N(n-m)})$. The matrix A_B consists of those columns of A corresponding to basic variables. The vector A_j is the j-th column of A.*
The reduced costs of a non-basic variable $x_{N(j)}$ are then defined as

$$c_{N(j)} - c_B A_B^{-1} A_{N(j)}.$$

Lemma 2.2. *Consider the linear program* (2.3). *A non-degenerate basic feasible solution x with basis B is an optimal solution of the linear program if and only if all reduced costs are non-negative*

$$c_{N(j)} - c_B A_B^{-1} A_{N(j)} \geq 0\ \forall j = 1, \ldots, n - m.$$

Proof. Follows directly from Theorem 2.2.1 of [12] and Theorem 4.1 of [3].
□ □

3 Algorithmic Idea and Quality Indicators of Sandwiching Algorithms

3.1 The Algorithmic Idea of Sandwiching Algorithms

Sandwiching algorithms as we will state them here are used to approximate the nondominated sets of bounded convex multiobjective optimization problems. They are not directly applicable to unbounded problems since for these problems, a polyhedral approximation of the Pareto front may not exist [25]. Recently, the Sandwiching algorithm idea has been extended to the approximation of unbounded Pareto fronts in [8] and [26].

The idea of the Sandwiching algorithm idea as described e.g. in [4], [16] or [24] is as follows. After an initial approximation, e.g. consisting of the extreme compromise solutions, has been computed, an inner and an outer approximation are constructed in the following way.

Let $\{z^1, \dots, z^n\}$ be the computed nondominated points. The inner approximation is then defined as the convex hull of the Pareto points, extended by the domination cone.

$$I^n := \operatorname{conv}\{z^1, \dots, z^n\} + \mathbb{R}^d_{\geq}.$$

The outer approximation is defined as the intersection of the half-spaces containing the Pareto front that support the Pareto points. Let $H(w^i, b^i) := \{z : (w^i)^T z = b^i\}$ be the supporting hyperplane of the Pareto front in z^i. Then the half-space $HS(w^i, b^i) := \{z : (w^i)^T z \leq b^i\}$ contains the Pareto front due to convexity. The outer approximation is defined as

$$O^n := \cap\{HS(w^i, b^i),\ i = 1, \dots, n\}.$$

Then, the approximation quality is determined, e.g. by computing the maximal epsilon indicator or polyhedral gauge distance between the inner and outer approximation. The facet of the inner approximation where the largest distance was measured is used as the starting point for computing a new nondominated point z^{n+1}: the facet's normal is used as the weighted sum parameter in the next scalarization problem so that the tangential hyperplane of the resulting weakly nondominated point is parallel to this facet of the former inner approximation.

Then, the inner approximation is updated by $I^{n+1} = \operatorname{conv}\{z^{n+1}, I^n\}$, the outer approximation is updated by $O^{n+1} := HS(w^{n+1}, b^{n+1}) \cap O^n$. If the approximation quality of the updated inner and outer approximation is not sufficient yet, a new scalarization problem is defined to compute the next nondominated point.

3.2 Common Quality Measures: Epsilon Indicator and Polyhedral Gauge

Many measures which are commonly used to assess the approximation quality of Sandwiching algorithms are based on the Hausdorff metric with different metrics used to compute the distance between elements of the respective sets. One of these criteria commonly used in Sandwiching algorithms (e.g. [4], [21]) but also other Pareto approximation algorithms or studies in multiobjective optimization, e.g. in [7], [18] or [27] is the *epsilon indicator* or *ϵ-indicator*.

The epsilon indicator $\delta^\epsilon(I, O)$ of a Sandwiching approximation I, O is the smallest number $\epsilon \geq 0$ such that for every $z \in O$ there exists a point in the inner approximation $z' \in I$ such that $z' \leq z + \epsilon \cdot e$ where $e = (1, \dots, 1) \in \mathbb{R}^d$

(Definitions 1 and 2 of [7]). To determine the epsilon indicator between an inner and outer Sandwiching approximation, it suffices to determine the epsilon indicator between vertices of the outer approximation and the inner approximation (Proposition 4.1 of [4]). The inner approximation of the Pareto front can be expressed as $P^T\xi + Q^T\mu$ with $\sum \xi = 1$ and $\xi, \mu \geq 0$ where the matrix P contains the nondominated points as columns and the matrix Q consists of the extreme rays of the domination cone $\mathcal{C}$ as columns. To calculate the ϵ-indicator, the extreme point s of the outer approximation is shifted in direction e until it reaches the inner approximation.

A linear program calculating the epsilon indicator distance from one extreme point s to the inner approximation is given by (Section 2.2.2 of [16])

$$\begin{aligned} \min_{\lambda,\mu,\eta} \ & (0,0,1) \cdot (\lambda, \mu, \eta)^T && (\epsilon_{LP}(s, I^n)) \\ \text{s.t.} \ & \begin{pmatrix} P^T & Q^T & -e \\ e^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \\ \eta \end{pmatrix} = \begin{pmatrix} s \\ 1 \end{pmatrix} \\ & \lambda, \mu, \eta \geq 0. \end{aligned}$$

Another metric that has been used to measure the approximation quality of the Sandwiching approximation, for example in [15] and [24], is based on the *polyhedral gauge*. W.l.o.g., we translate the bounded Pareto front such that the nadir point lies in the origin. For I^n, we define the reflected inner approximation as the reflection set of I^n (see Definition 2 of [23])

$$I^n_R := R(I^n) := \bigcup_{z \in I^n} \left\{ w \in \mathbb{R}^d : |w_i| = |z_i| \ \forall i = 1, \ldots, d \right\}.$$

For a vertex s of the outer approximation, the polyhedral gauge

$$\gamma_{I^n_R}(s) := \min \{\lambda \geq 0 : \ z \in \lambda I^n_R\}$$

(Definition 2.1.1 of [15]) describes the factor by which I^n_R has to be scaled to reach s. Then $\gamma_{I^n_R}$ is a norm (Lemma 6 of [23]). Since all Pareto points and s lie in the same orthant $\mathbb{R}^d_{\leq}$, we can omit reflecting I^n and work with it directly.

To obtain a linear program, the inverse of the polyhedral gauge is computed by scaling s by a factor α until it reaches the inner approximation. As in $\epsilon_{LP}(s, I^n)$, the matrix P contains the nondominated points as columns and the matrix Q consists of the extreme rays of the domination cone $\mathcal{C}$ as columns. The formulation as a linear program is given in Equation 2.4 of

[24]:

$$
\begin{aligned}
&\min_{\lambda,\mu,\alpha}\ (0,0,-1)\cdot(\lambda,\mu,\alpha)^T && (\gamma(s,I^n))\\
&\text{s.t. } \begin{pmatrix} P^T & Q^T & -s \\ e^T & 0 & 0 \end{pmatrix}\begin{pmatrix}\lambda\\ \mu\\ \alpha\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}\\
&\qquad \lambda,\mu,\alpha \geq 0.
\end{aligned}
$$

Thus, the polyhedral gauge can be calculated by solving the linear program $(\gamma(s,I^n))$ for every extreme point of the outer approximation s, obtaining the polyhedral gauge $\gamma^* = 1/\alpha^*$ and determining the maximal value. As a quality criterion we use the polyhedral gauge subtracted by one so that the quality value is zero for exact approximation.

4 Optimality Criteria for Epsilon Indicator and Polyhedral Gauge

Let the epsilon indicator or polyhedral gauge distance between a vertex s of the outer approximation and the inner approximation in iteration n of the Sandwiching algorithm be δ. If s is still a vertex of the updated outer approximation after adding a new Pareto point, under which conditions is its distance to the inner approximation still δ?

4.1 Checking whether the epsilon indicator has to be recalculated

In the following we assume that for the extreme point s, the solution of $(\epsilon_{LP}(s,I^n))$ is non-degenerate. The degenerate case is treated in Section 4.3.

Lemma 4.1. *Let us consider iteration $n+1$ of the Sandwiching algorithm. Let s be a vertex of both O^n and O^{n+1}, let (λ^*,μ^*,η^*) be a non-degenerate solution of $(\epsilon_{LP}(s,I^n))$ with basis B^*. The matrix A_{B^*} is defined by those columns of A corresponding to basic variables. Let $\bar{z}$ be the Pareto point that has been added to the Sandwiching approximation in iteration n. Then the following statements are equivalent:*

(i) (λ^*,μ^*,η^*) *solves* $(\epsilon_{LP}(s,I^{n+1}))$,

(ii) $r_{d+1}^T\cdot\begin{pmatrix}\bar{z}\\1\end{pmatrix}\leq 0$ *where* r_{d+1} *solves* $A_{B^*}^T\cdot r_{d+1}=e_{d+1}$.

Proof. For a non-degenerate solution $(\lambda^*, \mu^*, \eta^*)$ of $(\epsilon_{LP}(s, I^n))$, all reduced costs with respect to $(\lambda^*, \mu^*, \eta^*)$ and B^* are non-negative (Lemma 2.2). Thus, it holds for $N^* := \{1, \ldots, n\} \backslash B^*$ and $A_{N^*(j)}$ the $N^*(j)$-th column of A,

$$c_{N^*(j)} - c_{B^*} A_{B^*}^{-1} A_{N^*(j)} \geq 0 \ \forall j = 1, \ldots, n - (d+1). \tag{4.1}$$

At the end of iteration n, the new Pareto point $\bar{z}$ is added to the Sandwich approximation. The linear program $\epsilon_{LP}(s, I^n)$ is therefore updated to

$$\begin{aligned} \min_{\lambda, \lambda_{\text{new}}, \mu, \eta} \quad & (0, 0, 1) \cdot (\lambda, \lambda_{new}, \mu, \eta)^T \qquad\qquad (\epsilon_{LP}(s, I^{n+1})) \\ \text{s.t.} \quad & \begin{pmatrix} P^T & \bar{z} & Q^T & -e \\ e^T & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda & \lambda_{\text{new}} & \mu & \eta \end{pmatrix}^T = \begin{pmatrix} s \\ 1 \end{pmatrix} \end{aligned}$$

which only differs from $(\epsilon_{LP}(s, I^n))$ in one added column of the constraint matrix and one added variable λ_{new}.

We investigate under which condition $(\lambda^*, \mu^*, \eta^*)$ also solves $(\epsilon_{LP}(s, I^{n+1}))$. If for the feasible solution $(\lambda^*, \mu^*, \eta^*)$ of $(\epsilon_{LP}(s, I^{n+1}))$ the reduced costs for all non-basic variables $N := \{1, \ldots, n\} \backslash B^* = N^* \cup \lambda_{\text{new}}$ are non-negative,

$$c_{N(j)} - c_{B^*} A_{B^*}^{-1} A_{(j)} \geq 0 \ \forall j = 1, \ldots, n - (d+2),$$

it is an optimal solution (Lemma 2.2). Since for the non-degenerate solution $(\lambda^*, \mu^*, \eta^*)$ the reduced costs for all variables in N^* are non-negative (Equation (4.1)), $(\lambda^*, \mu^*, \eta^*)$ solves $(\epsilon_{LP}(s, I^{n+1}))$ if and only if (Lemma (2.2))

$$c_{\lambda_{new}} - c_{B^*} A_{B^*}^{-1} A_{\lambda_{\text{new}}} \geq 0. \tag{4.2}$$

With $c_{\lambda_{new}} = 0$, $c_{B^*} = (0, \ldots, 0, 1)$ and $A_{\lambda_{\text{new}}} = (\bar{z}, 1)^T$, we reduce (4.2) to

$$(0, \ldots, 0, 1) \cdot A_{B^*}^{-1} \cdot \begin{pmatrix} \bar{z} \\ 1 \end{pmatrix} \leq 0.$$

The multiplication $(0, \ldots, 0, 1) \cdot A_{B^*}^{-1}$ means that we actually only need the last row of $A_{B^*}^{-1}$. Denote the rows of $A_{B^*}^{-1}$ as $r_1, \ldots, r_{d+1}$ and let $I_{d+1} \in \mathbb{R}^{(d+1)\times(d+1)}$ be the identity matrix. The inverse matrix is defined as

$$A_{B^*}^{-1} \cdot A_{B^*} = \begin{pmatrix} r_1^T, & r_2^T, & \ldots, & r_{d+1}^T \end{pmatrix}^T \cdot A_{B^*} = I_{d+1}.$$

We can thus calculate the last row of the inverse using a system of linear equations of size $d+1$ (where $(A_{B^*})_j$ denotes the j-th column of A_{B^*} and $e_{d+1} \in \mathbb{R}^{d+1}$ the $d+1$-th unit vector)

$$\begin{cases} r_{n+1}^T \cdot (A_{B^*})_1 = 0 \\ r_{n+1}^T \cdot (A_{B^*})_2 = 0 \\ \vdots \\ r_{n+1}^T \cdot (A_{B^*})_{d+1} = 1 \end{cases}$$

$$\iff A_{B^*}^T \cdot r_{d+1} = e_{d+1}.$$

Solving one system of linear equations instead of inverting a matrix of the same size usually requires less computational effort. □ □

4.2 Checking whether the polyhedral gauge has to be recalculated

Analogous to the criterion for the epsilon indicator developed in Lemma 4.1, we can derive a similar criterion for the polyhedral gauge.

Lemma 4.2. *Let us consider iteration $n+1$ of the Sandwiching algorithm. Let s be a vertex of both O^n and O^{n+1}, let $(\lambda^*, \mu^*, \alpha^*)$ be a non-degenerate solution of $(\gamma(s, I^n))$ with basis B^*. The matrix A_{B^*} is defined by those columns of A corresponding to basic variables. Let $\bar{z}$ be the Pareto point that has been added to the Sandwiching approximation in iteration n. Then the following statements are equivalent:*

(i) $(\lambda^*, \mu^*, \alpha^*)$ *solves* $(\gamma(s, I^{n+1}))$*,*

(ii) $-r_{d+1}^T \cdot \begin{pmatrix} \bar{z} \\ 1 \end{pmatrix} \leq 0$ *where* r_{d+1} *solves* $A_{B^*}^T \cdot r_{d+1} = e_{d+1}$*.*

Proof. Using the same proof idea as in Lemma 4.1, we can show that $(\lambda^*, \mu^*, \alpha^*)$ solves $(\gamma(s, I^{n+1}))$ if and only if (Lemma (2.2))

$$c_{\lambda_{new}} - c_{B^*} A_{B^*}^{-1} A_{\lambda_{new}} \geq 0. \tag{4.3}$$

With $c_{\lambda_{new}} = 0$, $c_{B^*} = (0, \ldots, 0, -1)$ and $A_{\lambda_{new}} = (\bar{z}, 1)^T$, we reduce (4.3) to

$$(0, \ldots, 0, -1) \cdot A_{B^*}^{-1} \cdot \begin{pmatrix} \bar{z} \\ 1 \end{pmatrix} \leq 0.$$

The multiplication $(0, \ldots, 0, -1) \cdot A_{B^*}^{-1}$ in the optimality condition means that we actually only need (the negative of) the last row of $A_{B^*}^{-1}$. Just like in the proof of Lemma 4.1, we can obtain the last row of the inverse r_{d+1} by solving the system of linear equations $A_{B^*}^T \cdot r_{d+1} = e_{d+1}$. □ □

4.3 Degenerate solutions

The polyhedral gauge and the epsilon indicator are computed similarly. A vertex of the outer approximation is projected to the inner approximation. For the polyhedral gauge, this projection happens by scaling it, in the case of the epsilon indicator the extreme point is shifted. The amount by which the extreme point has been scaled or shifted is then the measured distance.

Definition 4.3. *The face in which the projected extreme point touches the inner approximation is called the face* associated *with s by the epsilon indicator or the polyhedral gauge.*

The criterion introduced in Lemma 4.1 can only be evaluated if the solution of the linear program is non-degenerate. If the solution of the epsilon indicator or polyhedral gauge linear programs is degenerate, the projected extreme point lies on a lower-dimensional face of the inner approximation. In the bi-criteria case, this situation can not occur. In the tri-criteria case, this means that the projected extreme point touches an edge of one of the triangles forming the inner approximation, see Figure 2. After obtaining a degenerate solution, one strategy of computing a non-degenerate solution is to perturb s.

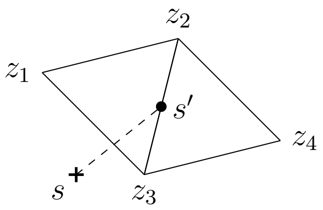


Figure 2: An approximation of a 3d Pareto front is given by points z_1, z_2, z_3, z_4. If the extreme point s is projected to the inner approximation in s' on the face z_2, z_3, the LP solution is degenerate. If it was projected onto the facet z_1, z_2, z_3 or z_2, z_3, z_4, the LP solution would be non-degenerate.

Remark 4.4. *In some rare cases, the basis matrix can be singular. In most cases, the matrix becomes regular by adding a small constant to the zero components of the extreme point. Otherwise, a least squares solution of the under-determined system of linear equations can be computed.*

5 Interpretation of the Criteria

5.1 Geometric Interpretation

In Lemma 4.1 and Lemma 4.2 we developed criteria that assess whether the epsilon indicator or the polyhedral gauge need to be recalculated for an extreme point s after a new point has been added to the approximation.

They can both be written as

$$\pm\, r_{d+1}^T \cdot \begin{pmatrix} \bar{z} \\ 1 \end{pmatrix} \leq 0 \tag{5.1}$$

where r_{d+1} solves $A_{B^*}^T \cdot r_{d+1} = e_{d+1}$

and $A_{B^*} \in \mathbb{R}^{(d+1)\times(d+1)}$ is of the form

$$A_{B^*} = \begin{pmatrix} p_1 & \cdots & p_i & q_1 & \cdots & q_j & -x \\ 1 & \cdots & 1 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

where $p_1, \ldots, p_i$ are Pareto points, $q_1, \ldots, q_j$ extreme rays of the domination cone with $i + j = d$. The criteria differ in the sign: positive for the epsilon indicator, negative for the polyhedral gauge, and in the definition of x with $x := e = (1, \ldots, 1)$ for the epsilon indicator, $x := s$ with the extreme point s for the polyhedral gauge.

We reformulate the criterion (5.1) so that it can easily be interpreted geometrically. To do this, we split the vector r_{d+1} into the vector $r_A := (r_{d+1}^1, \ldots, r_{d+1}^d)$ and the scalar $r_B := r_{d+1}^{d+1}$ which are calculated using the system of linear equations within (5.1) given by

$$\begin{pmatrix} p_1 & \ldots & p_i & q_1 & \ldots & q_j & -x \\ 1 & \ldots & 1 & 0 & \ldots & 0 & 0 \end{pmatrix}^T \cdot \begin{pmatrix} r_A \\ r_B \end{pmatrix} = \begin{pmatrix} 0 & \ldots & 0 & 1 \end{pmatrix}^T. \tag{5.2}$$

Using this notation, we can reformulate Equation (5.1) to

$$\pm r_{d+1} \cdot \begin{pmatrix} \bar{z} \\ 1 \end{pmatrix} \leq 0 \Longleftrightarrow \pm \frac{r_A}{|r_A|} \cdot \bar{z} \pm \frac{r_B}{|r_A|} \leq 0.$$

This equation looks like the Hesse normal form of a half-space. Then the criterion can be read as: the quality of extreme point s has to be recalculated if the new Pareto point $\bar{z}$ lies in the half-space

$$\left\{ \pm \frac{r_A}{|r_A|} \cdot x \pm \frac{r_B}{|r_A|} \leq 0,\ x \in \mathbb{R}^d \right\}. \tag{5.3}$$

In the following, we will interpret r_A and r_B with respect to the face associated with the extreme point s.

Lemma 5.1. *The vector r_A is perpendicular to the facet associated with the extreme point s by the epsilon indicator or polyhedral gauge.*

Proof. When we subtract the k-th rows, $k = 2, \ldots, i$ from the first row of the system of linear equations (5.2), we obtain $p_k \cdot r_A + r_B - (p_1 \cdot r_A +$

$r_B) = (p_k - p_1) \cdot r_A = 0$, so r_A is orthogonal to the (potentially lower-dimensional) facet spanned by the Pareto points $p_1 \dots, p_i$. Rows $j \dots, d$ of the system of equations are of the form $q_l \cdot r_A = 0$, $l = 1, \dots, j$, so r_A is perpendicular to the extreme cone rays forming the facet together with some Pareto points. □ □

Lemma 5.2. *The Euclidean distance between the facet associated with s by the epsilon indicator or polyhedral gauge and the origin is $\frac{r_B}{|r_A|}$.*

Proof. A facet touched by the projected extreme point is defined by at least one Pareto point, w.l.o.g. p_1. From Lemma 5.1 we know that r_A is a normal to the facet. The first row of (5.2) is $p_1 \cdot r_A + r_B = 0$. Thus, we can express the hyperplane defined by the facet as

$$\left\{x \in \mathbb{R}^d : r_A \cdot (x - p_1) = 0\right\} = \left\{x \in \mathbb{R}^d : \frac{r_A}{|r_A|} \cdot x + \frac{r_B}{|r_A|} = 0\right\}.$$

This is just the hyperplane defining the half-space in Equation (5.3). Since the Hesse normal form is defined by a unit normal and the distance of the hyperplane to the origin, $\frac{r_B}{|r_A|}$ is the value of this distance. □ □

Lemma 5.3. *The two developed criteria determining whether the quality has to be recalculated for an extreme point s, one for the polyhedral gauge (Lemma 4.2) and one for the epsilon indicator (Lemma 4.1), are the same.*

Proof. The criteria for the polyhedral gauge and the epsilon indicator (5.1) differ in the sign and the value of the vector x in one column of the matrix A_{B^*}. We note that the Lemmata 5.1 and 5.2 are independent of the value of x, so both half-spaces are defined by the same hyperplane.

It remains to show whether the normals point in the same direction. For an inner approximation given by Pareto points Z, the normals of facets of $\operatorname{conv}(Z) + \mathcal{C}$ can contain either only non-negative or non-positive components because our algorithm avoids dominated facets, see [16]. We write $a \geq 0$ if $a_i \geq 0 \ \forall i$. The last equation in the system of linear equations (5.2) is $-x \cdot r_A = 1$.

For the epsilon indicator, we have $x = e > 0$. To fulfil $-x \cdot r_A = 1$, r_A has to have at least one negative component, which means $r_A \leq 0$. Thus, in our minimization problem r_A is the outer normal of a facet of the inner approximation. For the polyhedral gauge, we have $x = s$. Since s dominates the Pareto front and the nadir point lies in the origin, $s \leq 0$. To fulfil $-x \cdot r_A = 1$, we need $r_A \geq 0$, therefore r_A is an inner normal of a facet of the inner approximation. Together with the different signs in Equation (5.3), we can see that the optimality criterion (5.1) forms the same half-space for both the epsilon indicator and the polyhedral gauge. □ □

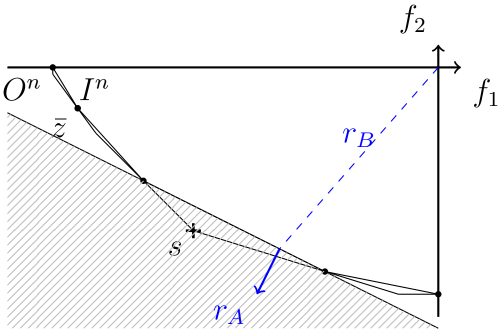


Figure 3: Geometric interpretation of the criterion Lemma 4.1 and 4.2 and Corollary 5.4: the distance of s to the inner approximation does not have to be recalculated if $\bar{z}$ lies outside of the shown half-space.

Corollary 5.4. *The epsilon indicator or polyhedral gauge quality of the extreme point s has to be recalculated if and only if the last computed nondominated point $\bar{z}$ lies inside the half-space defined by the facet associated with s and its outer normal. This situation is depicted in Figure 3.*

Proof. The statement follows directly from Lemmata 5.1, 5.2 and 5.3. □ □

Remark 5.5. *Although the criteria of epsilon indicator and polyhedral gauge can be interpreted as the same half-space, the result of the criterion evaluation may still be different since the basis matrix may differ.*

5.2 Comparison to the Criterion Developed by Dörfler et al. in [9]

In the article [9], a variant of the Sandwiching algorithm is introduced and analysed. As a concept of distance between the inner and outer approximation, the Hausdorff distance (defined using the Euclidean norm) is used. For the associated quadratic optimization problem, a criterion is derived using convex optimization theory which checks whether the Hausdorff distance to the inner approximation has to be recalculated for a vertex of the outer approximation. We state this criterion and compare it to the criteria developed for the epsilon indicator and polyhedral gauge in Lemmata 4.1 and 4.2.

The Euclidean distance between an extreme point s and the inner approximation can be computed by solving

$$\min \ ||p - s||_2 \text{ s.t. } p \in I^n. \qquad (QP(s, I^n))$$

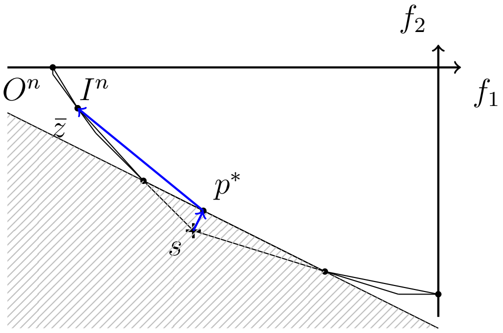


Figure 4: Geometric interpretation of Corollary 4.4 of [9]

The Hausdorff distance between the inner and outer Sandwiching approximation is given by $\delta^H(O^n, I^n) = ||p^* - s^*||$ (Corollary 4.2 of [9]) where s^* is the vertex of the outer approximation with the largest Euclidean distance to the inner approximation.

Lemma 5.6 (Corollary 4.4 of [9])**.** *Let us consider iteration $n+1$ of the Sandwiching algorithm. Let s be a vertex of both O^n and O^{n+1}, let p^* be a solution to $(QP(s, I^n))$ and $\bar{z}$ such that $I^{n+1} = \mathrm{cl}\,(\mathrm{conv}(I^n \cup \{\bar{z}\}))$. Then the following statements are equivalent:*

(i) p^* is a solution to $QP(s, I^{n+1})$,

(ii) $(p^* - s)^T(\bar{z} - p^*) \geq 0$.

Geometrically, the criterion given in Lemma 5.6 again forms a half-space, depicted in Figure 4: the distance of s to the inner approximation does not have to be recalculated if $\bar{z}$ lies outside of the shown half-space.

Corollary 5.7. *The criterion developed by Dörfler et al. in [9] for the Euclidean metric, developed from convex optimization techniques, is equivalent to the criteria that we developed for the epsilon indicator (Lemma 4.1) and the polyhedral gauge (Lemma 4.2) using techniques from linear optimization.*

This suggests that for other approximation quality indicators that could be used in Sandwiching, the same criterion may also determine whether the quality has to be recalculated.

6 Computing the Approximation Quality Efficiently in Sandwiching Algorithms

Previously, the Sandwiching algorithm required solving a linear program for each vertex of the outer approximation in each iteration of the algorithm to compute the epsilon indicator or polyhedral gauge quality. Using the results developed in this article, we can reduce the number of linear programs solved to compute the approximation quality, as implemented in Algorithm 1.

Algorithm 1 Finding the solution of LP(s, I^{n+1}), often avoiding solving the actual linear program

```
Require: Extreme point $s$, approximation ($I^{n+1}$, $O^{n+1}$),
         quality LP solutions from iteration $n$, target quality $\epsilon$
if $s$ was not an extreme point of $O^n$ then
    Solve LP($s, I^{n+1}$) and return its solution
Obtain the solution $(\lambda, \mu, \eta)$ of LP($s, I^n$) from memory
if $\eta \leq \epsilon$ then return $(\lambda, \mu, \eta)$
if $(\lambda, \mu, \eta)$ is degenerate then
    Perturb $s$ to $\tilde{s}$ and return the solution of LP($\tilde{s}, I^{n+1}$)
else
    if Criterion 4.1 (epsilon indicator) or 4.2 (polyhedral gauge) are fulfilled then
        return $(\lambda, \mu, \eta)$
    else
        Solve LP($s, I^{n+1}$) and return its solution
```

If information about neighbourhoods of the extreme points is available, a variant could be implemented that requires fewer evaluations of the criteria. We observe that due to the convexity of the Pareto front, the set of extreme points in the half-space must be connected. Therefore, starting from the newly created extreme points, we could evaluate the criterion for their neighbours. If the quality of an extreme point has to be recalculated, the criterion is also checked for its neighbours until there are no more neighbours left whose quality would have to be recalculated.

7 How Many Quality Calculations are Actually Necessary?

After demonstrating that using the criteria developed in this article it is possible to avoid solving many quality LPs, the question arises how many linear programs actually need to be solved.

Bi-criteria problems

Lemma 7.1. *In bi-criteria Sandwiching approximations, no extreme point changes its quality value after a Pareto point has been added to the approximation.*

Proof. Let a facet of the inner approximation be given by points p_1 and p_2, forming a triangle with the intersection s of their tangents. Let $\bar{z}$ be added to the approximation using a weight which is not the normal n of the facet p_1, p_2. The quality of s only has to be recomputed if $\bar{z}$ lies in the half-space $n \cdot y + d > 0,\ y \in \mathbb{R}^2$ with d the distance between the facet and the origin (Lemmata 4.1, 4.2, 5.6). Therefore, $\bar{z}$ would need to lie in the triangle p_1, p_2, s. But due to the convexity of the Pareto front, $\bar{z}$ would then need to have a tangent that is w.l.o.g. steeper than that of p_1 and flatter than that of p_2, which contradicts $\bar{z}$ not being created using facet p_1, p_2. □ □

Lemma 7.2. *Let the Pareto front of a bounded convex bi-objective optimization problem be approximated using the Sandwiching algorithm, let the two extreme compromises already be computed. Then in every iteration, the quality has to be recalculated for two extreme points.*

Proof. After a Pareto point $\bar{z}$ has been added, its tangent is added to the outer approximation. Since the extreme compromises have already been computed, the tangent in $\bar{z}$ will cut the tangents of two other Pareto points resulting in two additional extreme points. The extreme point that has been used to determine the new Pareto point is cut off. In Lemma 7.1 we showed that no existing extreme point will change its quality when a Pareto point is added to the approximation. Thus, only the quality of the two new extreme points has to be determined. □ □

Since the two necessary quality calculations are just those defined by the two new outer extreme points created when updating the outer approximation using the tangent of the new Pareto point $\bar{z}$, it is not necessary to evaluate the criteria of Lemmata 4.1, 4.2 or Corollary 4.4 of [9] at all. Implementing this method would lead to another speed-up for bi-criteria problems compared to the results illustrated in Section 8.

Arbitrary number of dimensions

We investigate for higher-dimensional problems whether a neighbourhood of facets can be defined so that only the qualities of those outer extreme points associated with the facets in this neighbourhood need to be checked for recomputation of the approximation quality.

This problem can alternatively be interpreted by facets of convex hulls using Corollary 5.4 together with Theorem 2.1 of [2]: After a Pareto point $\bar{z}$ has been added to the approximation given by Pareto points P, the approximation quality (epsilon indicator, polyhedral gauge or Euclidean metric) with respect to the extreme point s has to be recalculated if and only if the facet associated with s is not a facet of $\text{conv}(\bar{z} \cup P)$, i.e. if the facet associated with s is not a facet of the updated inner approximation polytope any more.

It is easy to verify that for tri-criteria problems, the facet f that $\bar{z}$ is associated with as well as those facets sharing an edge with f may be removed when updating the convex hull. The following example demonstrates that it is also possible that facets that only share a vertex with f may be removed.

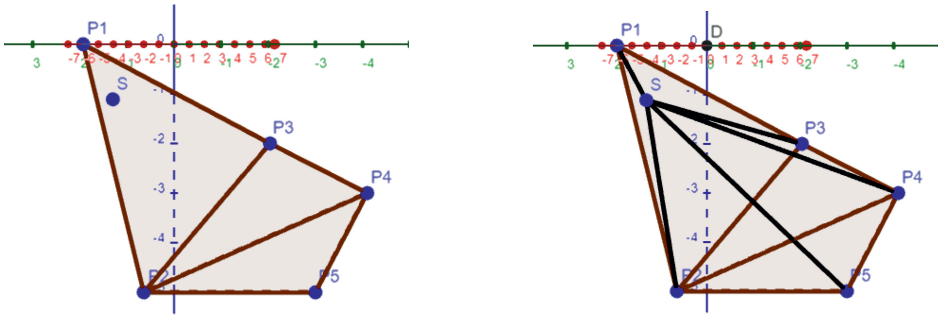
Let us consider a Pareto front that contains the following five Pareto points $P_1 = (-6, 0, 0)$, $P_2 = (-2, 0, -5)$, $P_3 = (-3, -3, -2)$, $P_4 = (-1.25, -4.5, -3)$, $P_5 = (0, -3, -5)$, the nadir point lies in the origin. The unit normal of the Pareto front in p is $p/\|p\|_2$. The facets of the inner approximation are given by the triangles t_1 with vertices $P_1P_2P_3$, t_2 with vertices $P_2P_3P_4$ and t_3 with vertices $P_2P_4P_5$. Facet t_2 shares an edge with t_1 while facet t_3 only shares a vertex with t_1. The point $S = (-6, -0.63, -1.11)$ lies between the inner and the outer approximation. One could define a multiobjective optimization problem such that S is the next Pareto point computed by the Sandwiching algorithm. We use the polyhedral gauge as a quality indicator. Then, S is associated with facet t_1. The point S lies inside the half-spaces defined by all three facets t_1, t_2, t_3 and their outer normals. This means that when S is added to the inner approximation, the facets t_1, t_2 and t_3 will be removed and the qualities of all outer extreme points associated with one of these facets may change. This example is illustrated in Figure 5.

In general, the number of facets of a convex hull that have at least one vertex in common with a facet is unbounded. Therefore, it is not possible to specify a maximal number of facets so that only those outer extreme points associated with these facets need to be checked for changing quality LP values.

8 Numerical Examples

We illustrate the saved number of linear programs and the improved total computation time using the approximation of d-dimensional spheres. We formulate the multiobjective optimization problem

$$\min \; f(x) = x \;\; \text{s.t.} \; x^T x \leq 1$$



(a) The inner approximation is formed by three facets. The new Pareto point S lies between the inner and outer approximation.

(b) The lines from S to all points lies outside of the inner approximation, i.e. all three facets will vanish when the convex hull of the Pareto points $P_1, \dots, P_5, S$ is computed.

Figure 5: Example of a Pareto front approximation using five points $P_1, \dots, P_5$ in three-dimensional objective space.

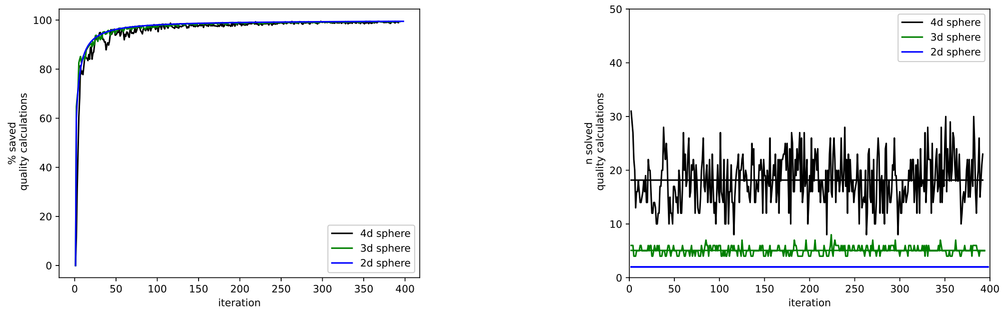
with $f : \mathbb{R}^d \to \mathbb{R}^d$, $x \in \mathbb{R}^d$. The Pareto front is given by the part of a d-dimensional unit sphere centred around the origin that intersects $\mathbb{R}^d_{\leq}$.

The following tests were performed on a laptop with 16 GB of RAM and an Intel Core i7-8665U processor with a clock rate of 1.9 GHz and 4 cores. The Sandwiching algorithm has been implemented in C++. We will use the epsilon indicator as the quality indicator.

We first approximate spherical Pareto fronts with 2, 3 and 4 objective dimensions by 400 Pareto points and apply Algorithm 1 to compute the approximation quality efficiently. We observe in Figure 6a that the percentage of saved LPs increases with the number of iterations and approaches 100 %. This illustrates that only the quality calculations of a small environment of outer extreme points near the new Pareto point are affected in every iteration.

Figure 6b documents the number of linear programs that remain to be solved in every iteration. In the bi-criteria case, exactly two linear programs are solved in every iteration which illustrates Lemma 7.2. For the three-dimensional sphere, between 4 and 8 linear programs are solved in every iteration. In the four-dimensional case, this number fluctuates between 8 and 31. While the behaviour gets more irregular for an increasing number of objective functions, the data suggests that the number of solved linear programs does not increase with the number of iterations.

The resulting decrease in approximation time is shown in Figure 7a, exemplary for a four-dimensional sphere. We can observe that the Sandwiching approximation time increases slower with the number of iterations if Algo-



(a) Percentage of saved quality calculation LPs

(b) Number of remaining LPs to be solved, together with their mean value

Figure 6: The effect on the number of computed quality calculation LPs by incorporating Lemma 4.1 into the Sandwiching algorithm when approximating spheres of different dimensions.

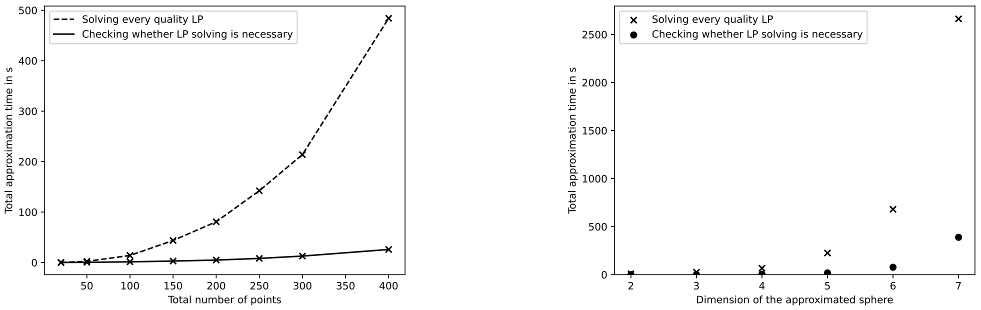
rithm 1 is applied to compute the approximation quality. E.g., the computation time of an approximation of a 4d sphere by 400 points only takes 27 s after incorporating the criterion versus 488 s when solving every quality LP.

Next, we approximate spheres with 2 to 7 objectives by 200 points, once solving a quality LP for every vertex of the outer approximation in every iteration and once using Algorithm 1. The percentage of saved LP runs shown in Figure 8b decreases from 98 % for the 2d sphere to 90 % for the 7d sphere. This can partly be explained by an increase in degenerate approximation quality LP solutions in higher dimensions. While we do not encounter degeneracy in the bi-criteria case and for 3,4 and 5 criteria less than 1 % of the LP solutions are degenerate, for 7 criteria there are 6 % degenerate LP solutions.

The approximation times are documented in Figure 7b. We observe that the approximation times increase much slower with the number of objective space dimensions if Algorithm 1 is applied to compute the approximation quality. The saved time percentage lies between 85 % and 94 % (see Figure 8a).

9 Conclusions

We developed an algorithm to compute the epsilon indicator and polyhedral gauge quality of an inner and outer polyhedral approximation more efficiently. It assesses whether the distance between a vertex of the outer



(a) Approximation of a four-dimensional sphere using 400 non-dominated points

(b) Approximation of spheres of different dimensions using 200 points each

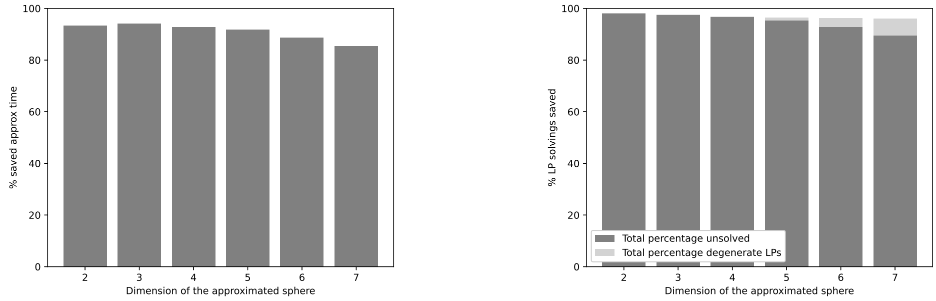
Figure 7: Sandwiching approximation times with and without Algorithm 1.

approximation and the inner approximation needs to be recalculated after a point has been added to the approximation.

By interpreting the criteria geometrically as a half-space, we could show that the criteria for the polyhedral gauge and epsilon indicator are equivalent. Moreover, they are also equivalent to a criterion developed by Dörfler et al. for the Euclidean norm using convex optimization theory. Therefore, the additional structure of a linear optimization problem given by the polyhedral gauge and epsilon indicator does not improve the result. Additionally, having obtained equivalent criteria using two different techniques for three different quality indicators, suggests that the same criterion can also be applied to other quality indicators that are used in Sandwiching algorithms.

For the bi-criteria case we showed that only 2 LPs need to be solved per iteration to compute the approximation quality. For more objectives, we demonstrated that no constant bound on the number of linear programs to be checked can be proved. However, examples suggest that the number of LPs remaining to be solved in every iteration does not increase with the number of points.

Incorporating the improved computation of the approximation algorithm into the Sandwiching algorithm greatly decreases the number of linear programs to be solved to compute the approximation quality. In numerical examples we see a decrease in approximation time of up to 94 %. Additionally, the approximation time increases more slowly with the number of iterations and the number of objective space dimensions.



(a) Percentage of approximation time that is saved by incorporating the criterion of Lemma 4.1.

(b) Percentage of quality LPs that do not need to be solved using Algorithm 1 and the percentage of degenerate LP solutions.

Figure 8: Comparing the results when solving every quality LP and when Algorithm 1 is applied for spheres of different dimensions, and 200 points each.

References

[1] Çağın Ararat, Firdevs Ulus, and Muhammad Umer. A norm minimization-based convex vector optimization algorithm. *Journal of Optimization Theory and Applications*, 194(2):681–712, 2022.

[2] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[3] Mokhtar S Bazaraa, John J Jarvis, and Hanis D Sherali. *Linear programming and network flows.* John Wiley & Sons, 2008.

[4] Rasmus Bokrantz and Anders Forsgren. An algorithm for approximating convex pareto surfaces based on dual techniques. *INFORMS Journal on Computing*, 25(2):377–393, 2013.

[5] Guillermo Cabrera-Guerrero, Matthias Ehrgott, Andrew J Mason, and Andrea Raith. Bi-objective optimisation over a set of convex subproblems. *Annals of Operations Research*, pages 1–26, 2021.

[6] David L Craft, Tarek F Halabi, Helen A Shih, and Thomas R Bortfeld. Approximating convex pareto surfaces in multiobjective radiotherapy planning. *Medical physics*, 33(9):3399–3407, 2006.

[7] Erik Diessel. Precise control of approximation quality in multicriteria optimization. Preprint, http://www.optimization-online.org/DB_HTML/2020/09/8036.html, 2021.

[8] Daniel Dörfler. On the approximation of unbounded convex sets by polyhedra. *Journal of Optimization Theory and Applications*, 194(1):265–287, 2022.

[9] Daniel Dörfler, Andreas Löhne, Christopher Schneider, and Benjamin Weißing. A benson-type algorithm for bounded convex vector optimization problems with vertex selection. *Optimization Methods and Software*, pages 1–21, 2022.

[10] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.

[11] Matthias Ehrgott, Lizhen Shao, and Anita Schöbel. An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50(3):397–416, 2011.

[12] Horst W Hamacher and Kathrin Klamroth. *Linear and network optimization*. Springer-Verlag, 2006.

[13] George K Kamenev. An algorithm for approximating polyhedra (in russian). *Computational mathematics and mathematical physics*, 4(36):533–544, 1996.

[14] George K Kamenev. Self-dual adaptive algorithms for polyhedral approximation of convex bodies. *Computational mathematics and mathematical physics*, 8(43):1073–1086, 2003.

[15] Kathrin Klamroth, Jørgen Tind, and Margaret M Wiecek. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56(3):413–437, 2003.

[16] Ina Lammel. *Approximation of Pareto Surfaces in Multi-Criteria Optimization*. PhD thesis, RPTU Kaiserslautern, 2023.

[17] Ina Lammel, Karl-Heinz Küfer, and Philipp Süss. An approximation algorithm for multi-objective mixed-integer convex optimization, 2023. Preprint at https://optimization-online.org/?p=22340.

[18] Julien Legriel, Colas Le Guernic, Scott Cotton, and Oded Maler. Approximating the pareto front of multi-criteria optimization problems. In

International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pages 69–83. Springer, 2010.

[19] Andreas Löhne, Birgit Rudloff, and Firdevs Ulus. Primal and dual approximation algorithms for convex vector optimization problems. *Journal of Global Optimization*, 60(4):713–736, 2014.

[20] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.

[21] Gijs Rennen, Edwin R van Dam, and Dick den Hertog. Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4):493–517, 2011.

[22] Günter Rote. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing*, 48(3):337–361, 1992.

[23] Bernd Schandl, Kathrin Klamroth, and Margaret M Wiecek. Introducing oblique norms into multiple criteria programming. *Journal of Global Optimization*, 23(1):81–97, 2002.

[24] Jorge Ivan Serna Hernandez. *Multi-objective optimization in Mixed Integer Problems with application to the Beam Selection Optimization Problem in IMRT.* PhD thesis, TU Kaiserslautern, 2011.

[25] Firdevs Ulus. Tractability of convex vector optimization problems in the sense of polyhedral approximations. *Journal of Global Optimization*, 72(4):731–742, 2018.

[26] Andrea Wagner, Firdevs Ulus, Birgit Rudloff, Gabriela Kováčová, and Niklas Hey. Algorithms to solve unbounded convex vector optimization problems. *arXiv preprint arXiv:2207.03200*, 2022.

[27] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.