SIAM J. OPTIM.
Vol. 29, No. 1, pp. 794–821

A BRANCH–AND–BOUND-BASED ALGORITHM FOR NONCONVEX MULTIOBJECTIVE OPTIMIZATION[*]

JULIA NIEBLING[†] AND GABRIELE EICHFELDER[†]

Abstract. A new branch–and–bound-based algorithm for smooth nonconvex multiobjective optimization problems with convex constraints is presented. The algorithm computes a so-called (ε, δ)-efficient set of all globally optimal solutions. We introduce the algorithm, which uses selection rules, discarding, and termination tests. The discarding tests are the most important aspect, as they examine in different ways whether a box can contain optimal solutions and determine by that the speed and effectiveness of the algorithm. We present a discarding test which combines techniques from the αBB method from global single objective optimization with outer approximation techniques from multiobjective convex optimization and the concept of local upper bounds from combinatorial multiobjective optimization. We apply the algorithm to several test instances as well as to an application in Lorentz force velocimetry.

Key words. multiobjective optimization, nonconvex optimization, global optimization, branch-and-bound algorithm, αBB method

AMS subject classifications. 90C26, 90C29, 90C30

DOI. 10.1137/18M1169680

1. Introduction.

For multiobjective optimization problems (MOPs) a variety of algorithms exist [5, 14, 16, 33]. Such problems appear in engineering or economics whenever various objective functions have to be minimized simultaneously. In general there is no point which minimizes all objective functions at the same time. Thus, one uses another optimality concept than the one in single objective optimization. However, just like in single objective optimization, we have to distinguish between locally and globally optimal solutions. While locally optimal solutions are only optimal in a neighborhood, globally optimal solutions are optimal on the whole feasible set. Most algorithms for multiobjective optimization problems aim at finding locally optimal solutions and are thus only appropriate for convex problems.

For solving multiobjective (convex) optimization problems several algorithms already exist; see, for example, [12, 23]. Most of these algorithms are based on scalarization approaches [14, 16, 18, 35]. In these approaches a new single objective problem depending on some parameters is formulated and solved by known methods for single objective optimization problems. With a set of parameter values and more iterations an approximation of the optimal solution set can be obtained. If we use only local methods to find optimal solutions of the nonconvex single objective problems, we will obtain just locally optimal solutions of the original vector-valued problem. If we use a global solver for each choice of parameters for the scalarization problems, this is a very time-consuming and inefficient approach. Moreover, most scalarizations turn some of the nonconvex objective functions into constraints, but it is well known that nonconvex constraints are especially difficult to handle [25]. For instance the weighted

[*] Received by the editors February 8, 2018; accepted for publication (in revised form) December 18, 2018; published electronically March 21, 2019.
http://www.siam.org/journals/siopt/29-1/M116968.html
Funding: The first author thanks Thuringian State Graduate Support Regulation, Carl-Zeiss-Stiftung, and DFG-funded Research Training Group 1567 “Lorentz Force Velocimetry and Lorentz Force Eddy Current Testing” for financial support.

[†] Institute for Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany (julia.niebling@tu-ilmenau.de, gabriele.eichfelder@tu-ilmenau.de).

sum scalarization avoids this, but it is also well known that this is not an appropriate scalarization for nonconvex problems. For that reason the development of global solvers for MOPs without using scalarizations is important. For an introduction to global optimization, see, for example, [21].

Many methods for global optimization use stochastic strategies. Commonly used algorithms are evolutionary algorithms; see, for example, [9, 10, 17, 43]. These algorithms try to find globally optimal solutions by way of mutation and crossover of individuals of a constructed population, i.e., some feasible points of the optimization problem. In fact, these procedures are able to find a global minimum in an infinite amount of time, but they do not guarantee finding a good solution in a finite time. That is one reason why it is of special interest to propose deterministic algorithms.

An approach to find globally optimal solutions for MOPs deterministically was introduced by Jahn [22]. There, a search region in the preimage space is discretized and then refined into "promising" areas. In higher dimensions of the preimage space, i.e., $n \geq 3$, a large number of function evaluations is required. This is due to the fact that Jahn's method does not use any information about derivatives. Another derivative-free algorithm for MOPs was proposed by Custódio and Madeira in [8] which is based on a direct search approach with a clever multistart strategy and is also able to find globally optimal solutions.

Other global optimization algorithms are based on branch-and-bound methods; see, for example, [1, 2, 7, 11, 20, 34, 42, 44]. For single objective optimization problems, two of the most well-known algorithms, which use box partitions, are the DIRECT algorithm [24] and the αBB method [34]. The DIRECT algorithm focuses on selecting boxes to have a good balance between a local and a global search strategy. However, it cannot guarantee finding good solutions in finite time [30]. In contrast to this, the αBB algorithm uses lower bounds of the global minimum, which are obtained by minimizing a convex underestimator of the objective function. These bounds are then improved until a given accuracy is reached. Other methods for finding lower bounds for global minima use maximal values of the dual problem [11], or use the Lipschitz constant [29, 45]. The DIRECT algorithm was also extended to MOPs; see, for example, [7, 44]. However, in most cases the multiobjective version of the DIRECT algorithm shows bad convergence results and has to be accelerated by another global or local optimization method.

The first branch–and–bound-based algorithm for more than one objective function and with some basic convergence results was introduced by Fernández and Tóth in [15]. The described procedure is for biobjective optimization problems. Recently (see [45] and also [37]) another algorithm for biobjective problems was proposed by A. Žilinskas and J. Žilinskas. They use the Lipschitz property of the objective functions and iterative trisections of the feasible set, which is assumed to be a box. An extension to more general constraints might be possible, though it is not apparent how to accomplish this.

Analogously to [15], we base our algorithm on a branch-and-bound approach. However, we give an algorithm for an arbitrary number of objective functions and we provide new discarding tests based on the concept of convex underestimators from the αBB method [34]. This results in convex MOPs which have to be considered on each subbox. We give improved lower bounds for them by using approaches from multiobjective convex optimization. We combine those with the concept of the so-called *local upper bounds* to obtain a new discarding test. The local upper bounds are a versatile concept from multiobjective combinatorial optimization. Finally, we are able to prove that we find an approximation of the set of globally optimal solutions for MOPs with predefined quality in finite time.

This paper starts with the basics of multiobjective and global optimization in section 2. In section 3 we first recall a typical branch-and-bound algorithm for MOPs. Then we introduce the new algorithm. In section 4 we prove the correctness and termination of the proposed algorithm. The handling of constraints is the topic of section 5. Numerical results, also for an application problem from Lorentz force velocimetry, are presented in section 6. We end with a conclusion and an outlook in section 7.

2. Basics of multiobjective and global optimization.

In this section we introduce the basic definitions and concepts which we need for the new algorithm. We study the following MOP:

$$\text{(MOP)} \qquad \min_{x \in X^0} f(x) = (f_1(x), \dots, f_m(x))^T \text{ s.t. } g_r(x) \leq 0 \text{ for all } r = 1, \dots, p,$$

where $f_j \colon \mathbb{R}^n \to \mathbb{R},\ j = 1, \dots, m$, are twice continuously differentiable functions and $g_r \colon \mathbb{R}^n \to \mathbb{R},\ r = 1, \dots, p$, are continuously differentiable convex functions. We do not discuss the case of nonconvex constraints here. For those the same difficulties [25] as in the single objective case arise and we want to concentrate on the multiobjective aspects within this paper. We assume that at least one of the objective functions is nonconvex. The set $X^0 \subseteq \mathbb{R}^n$ is assumed to be a nonempty *box* (also called a hyper rectangle), i.e., $X^0 = \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\}$ with two points $\underline{x}, \overline{x} \in \mathbb{R}^n$. Note that we write $x \leq y$ if $x_i \leq y_i$ for all $i = 1, \dots, n$ and $x < y$ if $x_i < y_i$ for all $i = 1, \dots, n$. Moreover, the feasible set $M := \{x \in X^0 \mid g_r(x) \leq 0,\ r = 1, \dots, p\}$ is assumed to be nonempty. For an arbitrary set $A \subseteq M$ we define the image set of A under f by $f(A) := \{f(x) \in \mathbb{R}^m \mid x \in A\}$.

A common optimality concept in multiobjective optimization is *efficiency*: A point $x^* \in M$ is said to be *efficient* for (MOP) if there does not exist another $x \in M$ such that $f(x) \leq f(x^*)$ and $f(x) \neq f(x^*)$. The set of all efficient points is called the *efficient set* and is denoted by X_E. Hence, the set of globally optimal solutions of (MOP) is the efficient set. We say x^1 *dominates* x^2 if $x^1, x^2 \in M$, $f(x^1) \leq f(x^2)$, and $f(x^1) \neq f(x^2)$. We can define similar terms in the image space. Let $x^* \in M$. A point $y^* = f(x^*)$ is said to be *nondominated* for (MOP) if x^* is efficient for (MOP). The set of all nondominated points is called the *nondominated set*. We say y^1 *dominates* y^2 if $y^1, y^2 \in \mathbb{R}^m$, $y^1 \leq y^2$, and $y^1 \neq y^2$. Moreover, we say y^1 *strictly dominates* y^2 if $y^1, y^2 \in \mathbb{R}^m$ and $y^1 < y^2$.

The aim of the new algorithm is to find an (ε, δ)-efficient set $\mathcal{A}$ of (MOP), which is defined next. Let e denote the m-dimensional all-ones vector $(1, 1, \dots, 1)^T \in \mathbb{R}^m$ and let $\|\cdot\|$ denote the Euclidean norm.

Definition 2.1. *Let $\varepsilon \geq 0$ and $\delta \geq 0$ be given.*

(i) *A point $\bar{x} \in M$ is an ε-efficient point of* (MOP) *if there does not exist another $x \in M$ with $f(x) \leq f(\bar{x}) - \varepsilon e$ and $f(x) \neq f(\bar{x}) - \varepsilon e$.*

(ii) *A set $\mathcal{A} \subseteq M$ is an (ε, δ)-efficient set of* (MOP) *if every point of $\mathcal{A}$ is an ε-efficient point of* (MOP) *and if for all $x^* \in X_E$ there is an $\bar{x} \in \mathcal{A}$ with $\|\bar{x} - x^*\| \leq \delta$.*

This definition of ε-efficiency was introduced in an equivalent version in [28]. A more general concept for approximate solutions for vector optimization can be found in [19]. Furthermore, a slightly different definition of ε-efficiency is presented in [32], where different accuracies for the objective functions are allowed.

We use some ideas and concepts of interval arithmetic in our algorithm. For an introduction to interval analysis we refer the reader to [36]. The set of all n-dimensional real boxes will be denoted by $\mathbb{IR}^n$. The width of a box $X \in \mathbb{IR}^n$ is defined as $\omega(X) := \|\overline{x}-\underline{x}\|$. Note that interval arithmetic is only applicable to functions which can be formulated as a concatenation of basic arithmetic operations and elementary functions like sin, cos, tan, log, exp, etc. We have to assume that the objective functions f_j, $j = 1, \ldots, m$, are of such a type. Interval arithmetic is a common tool for calculating lower bounds of function values on a given box; see [15, 36]. Another way to calculate such lower bounds was proposed for scalar-valued functions in [34] under the name αBB and is based on the concept of convex underestimators. A *convex underestimator* for a function $h\colon \mathbb{R}^n \to \mathbb{R}$ on a box $X = [\underline{x}, \overline{x}] \in \mathbb{IR}^n$ is a convex function $\widehat{h}\colon X \to \mathbb{R}$ with $\widehat{h}(x) \leq h(x)$ for all $x \in X$. A convex underestimator of a twice continuously differentiable function h on X can be stated as

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

where $\alpha \geq \max\{0, -\min_{x\in X} \lambda_{\min}(x)\}$. Here, $\lambda_{\min}(x)$ denotes the smallest eigenvalue of the Hessian $H_h(x)$ of h in x [34]. A lower bound for $\lambda_{\min}(x)$ over X can be calculated with the help of interval arithmetic and Gerschgorin's theorem; see, e.g., [1]. See also [41] for improved lower bounds for $\lambda_{\min}(x)$. We use the MATLAB toolbox Intlab [38] to calculate these lower bounds.

Clearly, if $\tilde{X} \subseteq X$ and h_α is a convex underestimator of h on X, then h_α is also a convex underestimator of h on $\tilde{X}$. With our method to calculate α on X and $\tilde{\alpha}$ on $\tilde{X} \subseteq X$ we always obtain $\tilde{\alpha} \leq \alpha$. For simplicity of presentation, in this work we use only the parameter α for which h_α is a convex underestimator of h on the initial box $X = X^0$, even in the case in which we consider the function on a subbox of X^0. Furthermore, if we use the boundaries $\underline{\tilde{x}}$ and $\overline{\tilde{x}}$ of $\tilde{X} = [\underline{\tilde{x}}, \overline{\tilde{x}}] \subseteq X$ to define the convex underestimator $\tilde{h}_\alpha\colon \mathbb{R}^n \to \mathbb{R}$ on $\tilde{X}$ by $\tilde{h}_\alpha(x) := h(x) + \frac{\alpha}{2}(\underline{\tilde{x}} - x)^T(\overline{\tilde{x}} - x)$, we immediately obtain $\tilde{h}_\alpha(x) \geq h_\alpha(x)$ for all $x \in \tilde{X}$; see [34].

The main benefit of convex underestimators is that the minimum value of h_α over X, which can be calculated by standard techniques from convex optimization, delivers a lower bound for the values of h on X. There are also other possibilities for the calculation of convex underestimators. For example, in [1] special convex underestimators for bilinear, trilinear, fractional, fractional trilinear, or univariate concave functions were defined. Here, we restrict ourselves to the above proposed convex underestimator. The theoretical results remain true in the case in which the above underestimators are replaced by tighter ones. Another important benefit is stated in the following remark.

Remark 2.2 (see [34]). For all $\alpha \geq 0$ the maximal pointwise difference between h and h_α is $\frac{\alpha}{2}\omega(X)^2$, i.e., $\max_{x\in X} |h(x) - h_\alpha(x)| = \frac{\alpha}{2}\omega(X)^2$.

In the next lemma we show that the distance between the minimal value of a convex underestimator and the other function values of a smooth function h over a box is bounded by a given $\varepsilon > 0$ if the box width is small enough. This lemma is applied to f_j, $j = 1, \ldots, m$, later in this paper.

LEMMA 2.3. *Let a box $X \in \mathbb{IR}^n$, a twice continuously differentiable nonconvex function $h\colon \mathbb{R}^n \to \mathbb{R}$, a constant $\alpha \geq \max\{0, -\min_{x\in X} \lambda_{\min}(x)\}$, and a positive scalar $\varepsilon > 0$ be given. Moreover, let $L > 0$ be chosen such that $L \geq \sqrt{n}|\frac{\partial}{\partial x_i}h(x)|$ for all*

$i \in \{1, \ldots, n\}$ and $x \in X$. Let $\tilde{X} = [\underline{\tilde{x}}, \overline{\tilde{x}}]$ be a box with $\tilde{X} \subseteq X$ and with

$$\omega(\tilde{X}) \leq -\frac{L}{\alpha} + \sqrt{\frac{L^2}{\alpha^2} + \frac{\varepsilon}{\alpha}} =: \delta_X \tag{1}$$

and define $\tilde{h}_\alpha \colon \mathbb{R}^n \to \mathbb{R}$ by $\tilde{h}_\alpha(x) := h(x) + \frac{\alpha}{2}(\underline{\tilde{x}} - x)^T(\overline{\tilde{x}} - x)$, which is a convex underestimator of h on $\tilde{X}$. Then for $v := \min_{x \in \tilde{X}} \tilde{h}_\alpha(x)$ it holds that $|h(x) - v| \leq \frac{\varepsilon}{2}$ for all $x \in \tilde{X}$.

Proof. Note that $\alpha \neq 0$ because h is nonconvex. Let $\tilde{x}$ be a minimal solution of $\min_{x \in \tilde{X}} \tilde{h}_\alpha(x)$, i.e., $v = \tilde{h}_\alpha(\tilde{x})$. With Remark 2.2 it follows that $|h(\tilde{x}) - v| = |h(\tilde{x}) - \tilde{h}_\alpha(\tilde{x})| \leq \frac{\alpha}{2}\omega(\tilde{X})^2$. Let $x, y \in \tilde{X}$ be arbitrarily chosen now. By the mean value theorem there exists $\xi \in \{\lambda x + (1 - \lambda)y \in \mathbb{R}^n \mid \lambda \in (0, 1)\}$ with $h(x) - h(y) = \nabla h(\xi)^T(x - y)$. Together with the Cauchy–Schwarz inequality we obtain $|h(x) - h(y)| \leq \|\nabla h(\xi)\|\|x - y\|$. As

$$\|\nabla h(\xi)\| = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial}{\partial x_i} h(\xi)\right)^2} \leq \sqrt{\sum_{i=1}^{n} \frac{L^2}{n}} = L,$$

we derive $|h(x) - h(y)| \leq L\omega(\tilde{X})$ for all $x, y \in \tilde{X}$. Let $x \in \tilde{X}$ be arbitrarily chosen. Then, due to (1), it follows that for the distance between v and $h(x)$ we have

$$|h(x) - v| \leq |h(x) - h(\tilde{x})| + |h(\tilde{x}) - v| \leq L\omega(\tilde{X}) + \frac{\alpha}{2}\omega(\tilde{X})^2 \leq \frac{\varepsilon}{2}. \qquad \square$$

The constant L in the above lemma can be obtained by using techniques from interval arithmetic, for instance with the help of the MATLAB toolbox Intlab [38].

As we are considering vector-valued functions in this work, we use convex underestimators of every component function on its own. Thus, we denote the vector-valued convex underestimator of the function $f \colon \mathbb{R}^n \to \mathbb{R}^m$ by $f_\alpha \colon \mathbb{R}^n \to \mathbb{R}^m$, $x \mapsto (f_{1,\alpha}(x), \ldots, f_{m,\alpha}(x))^T$, where $f_{j,\alpha}$ is a convex underestimator of f_j, $j = 1, \ldots, m$. Note that we choose for the convex underestimators of each objective function the same parameter α for simplicity of presentation, although different ones are possible as well. Lemma 2.3 can easily be generalized to the vector-valued case by considering each objective function on its own.

3. The new branch–and–bound-based algorithm.

For a clearer presentation we introduce the algorithm for box-constained MOPs first. The handling of constraints will be discussed in section 5. Thus, in the following, we assume the MOP to be given as

$$\min_{x \in X^0} f(x) = (f_1(x), \ldots, f_m(x))^T, \tag{P}$$

i.e., $M = X^0$. Algorithm 1 gives a basic branch-and-bound algorithm, as already proposed in [15].

The lists $\mathcal{L}_W$ and $\mathcal{L}_S$ are the working list and the solution list, respectively. A typical selection rule is the one proposed in [15].

Selection rule. *Select the box $X^* \in \mathcal{L}_W$ with a minimum lower bound of f_1.*

In our algorithm this lower bound will be calculated by underestimating f_1 within the considered box by a convex underestimator. In [15] the lower bound is calculated by interval arithmetic. Certainly, it is possible to replace f_1 by any f_j, $j \in \{1, \ldots, m\}$,

Algorithm 1.

```
INPUT: $X^0 \in \mathbb{IR}^n, f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^m)$.
OUTPUT: $\mathcal{L}_S$.
$\mathcal{L}_W \leftarrow \{X^0\},\ \mathcal{L}_S \leftarrow \emptyset$;
while $\mathcal{L}_W \neq \emptyset$ do
    select a box $X^*$ from $\mathcal{L}_W$ and delete it from $\mathcal{L}_W$;        Selection rule
    bisect $X^*$ perpendicularly to a direction of maximum width $\rightarrow X^1, X^2$;
    for $l = 1, 2$ do
        if $X^l$ cannot be discarded then                                          Discarding tests
            if $X^l$ satisfies a termination rule then                             Termination rule
                store $X^l$ in $\mathcal{L}_S$;
            else store $X^l$ in $\mathcal{L}_W$.
```

or by a weighted sum of the objectives or similar. In [15] Fernández and Tóth use a similar termination rule to the one which is formulated next. Here, $F\colon \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ is the so-called *natural interval extension* of f, which evaluates the function with interval arithmetic; see [36]. For F it holds for any box X^* that $\{f(x) \in \mathbb{R}^m \mid x \in X^*\} \subseteq F(X^*)$.

Termination rule. *Store X^* in $\mathcal{L}_S$ if the following condition for given $\varepsilon, \delta > 0$ holds: $\omega(X^*) < \varepsilon$ and $\omega(F(X^*)) < \delta$.*

We use a much more detailed termination procedure, which we introduce in subsection 3.3. Our modified termination rule guarantees the (ε, δ)-efficiency of the calculated approximation set. For the discarding test several criteria have already been proposed in the literature. A first one is to use information on the monotonicity of the objective functions. Fernández and Tóth introduce in [15] a monotonicity test for biobjective optimization problems. A generalization to more than two objective functions can be found in [40]. Another class of discarding tests compares known objective values, which serve as upper bounds for the nondominated set, with lower bounds for the function values over the subboxes. We will follow this approach. Of course it is possible to combine the discarding tests in order to accelerate the algorithm and use the different advantages of each test. Nevertheless, in the next sections we focus on new ideas for a discarding test. We use the concept of convex underestimators to obtain multiobjective convex optimization problems. The nondominated set of these problems serve as lower bounds. To be able to compare these sets with upper bounds numerically we use outer approximations, which are polyhedral sets and which are lower bounds of the nondominated sets. We combine the new discarding test with a new termination procedure and test the new algorithm afterwards.

3.1. Upper bounds, and lower bounds by convex underestimators.

We will generate a *stable* set $\mathcal{L}_{PNS}$ of objectives values (called the *provisional nondominated set*) representing upper bounds for the global nondominated set for (P). A set $\mathcal{N} \subseteq \mathbb{R}^m$ is *stable* if for any $y^1, y^2 \in \mathcal{N}$ either $y^1 \nleq y^2$ or $y^1 = y^2$ holds. Every time when a point q is a new candidate for $\mathcal{L}_{PNS}$ we check if this point is dominated by any other point of $\mathcal{L}_{PNS}$. In this case q will not be included in $\mathcal{L}_{PNS}$. Otherwise, q will be added to $\mathcal{L}_{PNS}$ and all points dominated by q will be removed. Figure 1 shows an example for $\mathcal{L}_{PNS}$.

Having this list of upper bounds, a discarding test for a given box X^* also requires a lower bound for the values of f over X^*. Let $LB \subseteq \mathbb{R}^m$ be a set with $f(X^*) \subseteq LB + \mathbb{R}^m_+$. If $LB \subseteq (\mathcal{L}_{PNS} + \mathbb{R}^m_+) \setminus \mathcal{L}_{PNS}$, then the box X^* can be discarded, because

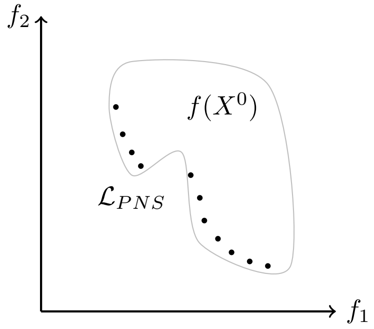


FIG. 1. *Example for $\mathcal{L}_{PNS}$, $m = 2$.*

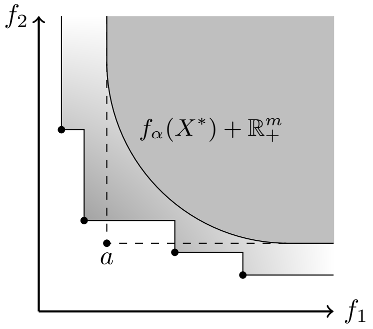


FIG. 2. *Upper image set $f_\alpha(X^*) + \mathbb{R}^m_+$ for $m = 2$, $\mathcal{L}_{PNS}$, and an ideal point a of f_α.*

every point of $f(X^*)$ is in $(\mathcal{L}_{PNS} + \mathbb{R}^m_+) \setminus \mathcal{L}_{PNS}$. Hence, every point of $f(X^*)$ is dominated by one point of $\mathcal{L}_{PNS}$. For the set LB we first recall the approach proposed so far in the literature, (I), and then present our new approach which consists of two steps, (II) and (III).

(I) Proposed in [15]: LB is chosen as the lower vertex of a box which contains all values of f on X^* and which is generated by the natural interval extension F of f on X^*; see [36].

(II) LB is chosen as the ideal point of the convex underestimators of the functions f_j over X^*. Recall that the *ideal point* of an MOP is determined componentwise by minimizing each objective function individually, i.e., we choose the point

$$a \in \mathbb{R}^m \quad \text{with} \quad a_j := \min\{f_{j,\alpha}(x) \mid x \in X^*\} \quad \text{for} \quad j = 1, \dots, m, \tag{2}$$

and set $LB = \{a = (a_1, \dots, a_m)\}$. For an illustration, see Figure 2.

(III) Find a tighter and not necessarily singleton set LB, i.e., a set LB with $f(X^*) \subseteq LB + \mathbb{R}^m_+$ and with $(LB + \mathbb{R}^m_+) \setminus (f(X^*) + \mathbb{R}^m_+)$ as small as possible by using convex underestimators and techniques from multiobjective convex optimization. We illustrate and discuss this new discarding test in subsection 3.2.

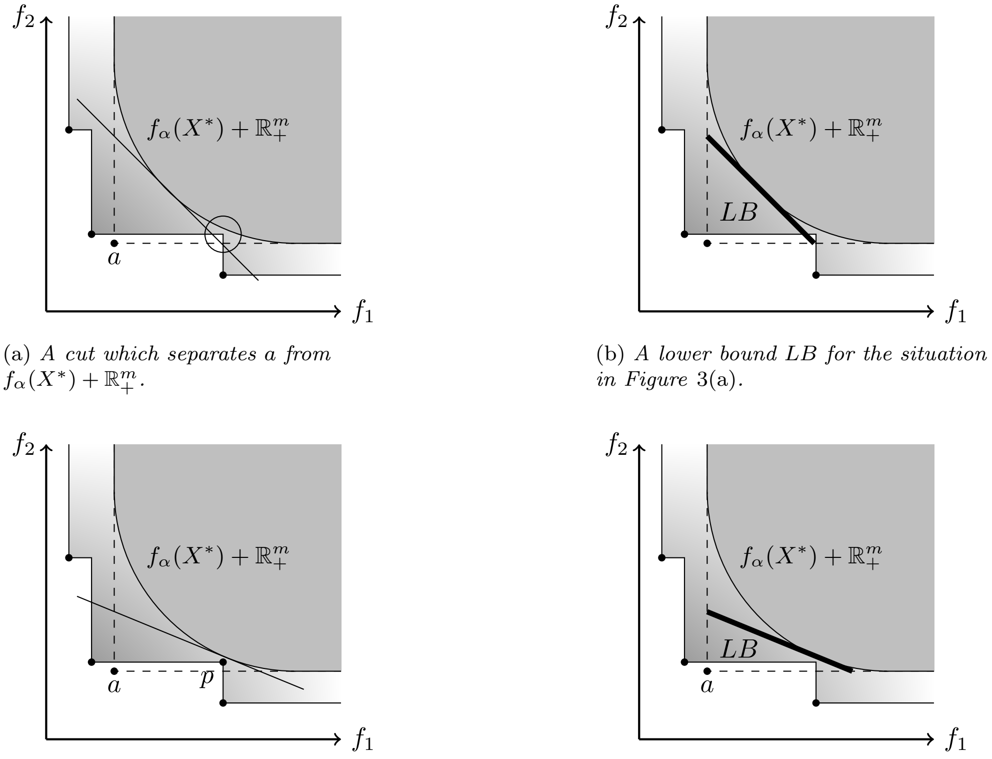
We start by briefly discussing the first step of our new approach, i.e., we show that a from (II) (see (2)) already delivers a lower bound.

LEMMA 3.1. *Let $f_{j,\alpha}$ be a convex underestimator of f_j on X^* for $j = 1, \dots, m$ and define $a \in \mathbb{R}^m$ by (2). Then $a \leq f(x)$ for all $x \in X^*$, i.e., $f(X^*) \subseteq \{a\} + \mathbb{R}^m_+$.*

Proof. Let $j \in \{1, \dots, m\}$. Because $f_{j,\alpha}$ is a convex underestimator of f_j on X^* and from the definition of a_j, it follows that $a_j \leq f_{j,\alpha}(x) \leq f_j(x)$ for all $x \in X^*$. □

Numerical experiments show that using (I) or (II) makes no big difference. In some cases the lower bounds by interval arithmetic are better than the ones by convex underestimators. In other cases, the converse holds. However, (II) has the advantage that the maximal error between the computed lower bound and actual function values is bounded (see Remark 2.2 and Lemma 2.3). Moreover, it is possible to improve (II) to (III) by using the convexity of f_α, which is not possible for (I). This is the basic idea of our new discarding test.

3.2. Improved lower bounds by selected Benson cuts. The tighter bounds will be reached by adding cuts as known from Benson's outer approximation algorithm for convex vector optimization problems; see [3, 13]. The cuts separate selected points p from the upper image set of the convex underestimators such that the cuts are supporting hyperplanes; see Figures 3(a) and 3(c). This leads to new sets LB, as can be seen in Figures 3(b) and 3(d). As we can see, the cut in Figure 3(a) would lead to a set LB which does not allow us to discard the box. This is due to the white triangle, which is circled in Figure 3(a). This triangle is not dominated by any point of $\mathcal{L}_{PNS}$ but it is in $LB + \mathbb{R}^2_+$. However, the cut in Figures 3(c) and 3(d) allows us to discard the box. First, we explain how to choose the points p such that we can generate cuts as in Figures 3(c) and 3(d). Then we discuss how to calculate the cuts in more detail.


f_2
$f_\alpha(X^*) + \mathbb{R}^m_+$
a
f_1
f_2
$f_\alpha(X^*) + \mathbb{R}^m_+$
LB
f_1
f_2
$f_\alpha(X^*) + \mathbb{R}^m_+$
a
p
f_1
f_2
$f_\alpha(X^*) + \mathbb{R}^m_+$
LB
a
f_1

(a) *A cut which separates* a *from* $f_\alpha(X^*) + \mathbb{R}^m_+$.

(b) *A lower bound* LB *for the situation in Figure* 3(a).

(c) *A cut which separates* p *from* $f_\alpha(X^*) + \mathbb{R}^m_+$.

(d) *A lower bound* LB *for the situation in Figure* 3(c).

FIG. 3. *Possible cuts to obtain a tighter set* LB *with* $f(X^*) \subseteq LB + \mathbb{R}^m_+,\ m = 2$.

We use as points p so-called local upper bounds, which is a versatile concept mainly used in multiobjective combinatorial optimization. Let a finite and stable list of function values $\mathcal{N} \subseteq f(X^0)$ be given. We also call points in $\mathcal{N}$ feasible points. Let $\hat{Z}$ be a box with $f(X^0) \subseteq \operatorname{int}(\hat{Z})$ (where int denotes the interior). The *search region* S is the set which contains all points which are not dominated by $\mathcal{N}$, i.e.,

$$S := \{z \in \operatorname{int}(\hat{Z}) \mid \forall q \in \mathcal{N},\ q \nleq z\} = \operatorname{int}(\hat{Z}) \setminus \left(\bigcup_{q \in \mathcal{N}} \{q\} + \mathbb{R}^m_+ \right). \tag{3}$$

This set can be characterized with the help of *local upper bounds*, which are the elements of the local upper bound set as defined below.

DEFINITION 3.2 (see [27]). *Let $\mathcal{N}$ be a finite and stable set of feasible points. A list $\mathcal{L} \subseteq \hat{Z}$ is called a local upper bound set with respect to $\mathcal{N}$ if*

(i) $\forall z \in S \ \exists p \in \mathcal{L} : z < p$,

(ii) $\forall z \in (\mathrm{int}(\hat{Z})) \setminus S \ \forall p \in \mathcal{L} : z \not< p$, *and*

(iii) $\forall p^1, p^2 \in \mathcal{L} : p^1 \not\lneq p^2$ *or* $p^1 = p^2$.

The next lemma gives an equivalent characterization [27].

LEMMA 3.3 (see [27]). *A set $\mathcal{L}$ is called a local upper bound set with respect to a finite and stable set of feasible points $\mathcal{N}$ if and only if $\mathcal{L}$ consists of all points $p \in \hat{Z}$ that satisfy the following two conditions:*

(i) *no point of $\mathcal{N}$ strictly dominates p; and*

(ii) *for any $z \in \hat{Z}$ such that $z \geq p$, $z \neq p$, there exists $\bar{z} \in \mathcal{N}$ such that $\bar{z} < z$, i.e., p is a maximal point with property* (i).

The following lemma is useful for understanding the relations between $\mathcal{N}$ and $\mathcal{L}$ and will be important for our proof of Lemma 3.5. It is due to Klamroth [26].

LEMMA 3.4. *Let $\mathcal{L}$ be a local upper bound set with respect to a finite and stable set of feasible points $\mathcal{N}$. For every $\bar{z} \in \mathcal{N}$ and for every $j \in \{1, \dots, m\}$ there is a $p \in \mathcal{L}$ with $\bar{z}_j = p_j$ and $\bar{z}_r < p_r$ for all $r \in \{1, \dots, m\} \setminus \{j\}$.*

Proof. Let $\bar{z} \in \mathcal{N} \subseteq \mathrm{int}(\hat{Z})$. As $\mathcal{N}$ is a finite set, there exists a neighborhood of $\bar{z}$ such that no other point of $\mathcal{N}$ is contained in that neighborhood. Hence, let $\nu > 0$ be such that $N_\nu(\bar{z}) := \{z \in \mathbb{R}^m \mid \|\bar{z} - z\| \leq \nu\} \subseteq \mathrm{int}(\hat{Z})$ and with $N_\nu(\bar{z}) \cap \mathcal{N} = \{\bar{z}\}$. Then we obtain for ν small enough $N_\nu(\bar{z}) \cap S = N_\nu(\bar{z}) \setminus (\{\bar{z}\} + \mathbb{R}^m_+)$. Fix now a $j \in \{1, \dots, m\}$ and let $(\delta_t)_{t \in \mathbb{N}}$ be a null sequence with $\nu > \delta_t > 0$ for all $t \in \mathbb{N}$. Then we consider the sequence $(q^t)_{t \in \mathbb{N}}$ defined componentwise by $q^t_j = \bar{z}_j - \delta_t$ and $q^t_r = \bar{z}_r$ for all $r \in \{1, \dots, m\} \setminus \{j\}$ and for all $t \in \mathbb{N}$. Then $\lim_{t \to \infty} q^t = \bar{z}$ and $q^t \in N_\nu(\bar{z}) \cap S$ for all $t \in \mathbb{N}$. Hence, with Definition 3.2(i) we conclude that there is a local upper bound $p^t \in \mathcal{L}$ with $q^t < p^t$ for every $t \in \mathbb{N}$. Since, $\mathcal{L}$ is a finite set, additionally the sequence $(p^t)_{t \in \mathbb{N}}$ contains a constant subsequence with a (limit) value $\bar{p}^j \in \mathcal{L}$. For its limit value it holds that $\bar{z} \leq \bar{p}^j$. Moreover, for all $t \in \mathbb{N}$ of the constant subsequence we have $\bar{z}_r = q^t_r < p^t_r = \bar{p}^j_r$ for all $r \in \{1, \dots, m\} \setminus \{j\}$. By Lemma 3.3(i) it follows that $\bar{z}_j = \bar{p}^j_j$. ◻

As a result of Lemma 3.4, for every $\bar{z} \in \mathcal{N}$ there exists a local upper bound $p \in \mathcal{L}$ with $\bar{z} \leq p$. In our context the list $\mathcal{L}_{PNS}$ is the list $\mathcal{N}$. The local upper bound set will be denoted by $\mathcal{L}_{LUB}$. An algorithm which calculates $\mathcal{L}_{LUB}$ w.r.t. $\mathcal{L}_{PNS}$ can be found in [27]. Figure 4 illustrates the sets $\mathcal{L}_{PNS}$ and $\mathcal{L}_{LUB}$. The values $\overline{M}_1$ and $\overline{M}_2$ in Figure 4 are upper bounds for the values of f_1 and f_2 in X^0, which do not have to be tight bounds and which can easily be computed, for example, with interval arithmetic.

The local upper bounds w.r.t. the set $\mathcal{L}_{PNS}$ are important as we can discard a box X^* if no local upper bound is contained in the set $f_\alpha(X^*) + \mathbb{R}^m_+$.

LEMMA 3.5. *Let a box $X^* \in \mathbb{IR}^n$, $X^* \subseteq X^0$, be given and let $f_{j,\alpha}$ be a convex underestimator of f_j on X^* for $j = 1, \dots, m$. Let $\mathcal{L}_{LUB} \subseteq \mathbb{R}^m$ be the local upper bound set w.r.t. $\mathcal{L}_{PNS}$. If*

$$\forall \bar{p} \in \mathcal{L}_{LUB} : \bar{p} \notin f_\alpha(X^*) + \mathbb{R}^m_+, \tag{4}$$

then X^* does not contain any efficient point of (P).

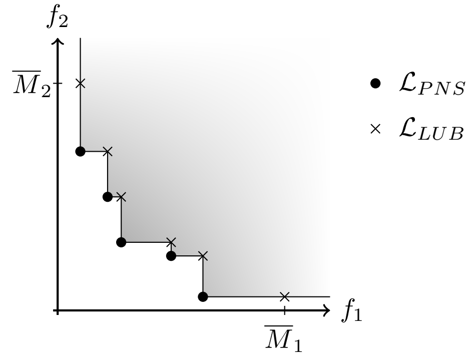


FIG. 4. *$\mathcal{L}_{PNS}$ and $\mathcal{L}_{LUB}$; cf.* [27, *Figure* 1].

Proof. Assume that there is some efficient point x^* of (P) with $x^* \in X^*$. Because $f_{j,\alpha}$ is a convex underestimator of f_j on X^* for all $j \in \{1,\dots,m\}$, no local upper bound $\bar{p} \in \mathcal{L}_{LUB}$ belongs to $f(X^*) + \mathbb{R}^m_+$. Thus, $f(x^*) \not\leq \bar{p}$ for all $\bar{p} \in \mathcal{L}_{LUB}$. From Definition 3.2(i) it follows that $f(x^*)$ cannot be an element of the search region S. With (3) we conclude that there exists a point $q \in \mathcal{L}_{PNS}$ with $q \leq f(x^*)$. As q is the image of a feasible point of (P) and as x^* is efficient for (P) we obtain $f(x^*) = q \in \mathcal{L}_{PNS}$. By Lemma 3.4, a local upper bound $p \in \mathcal{L}_{LUB}$ exists with $f(x^*) \leq p$ or, equivalently, $p \in \{f(x^*)\} + \mathbb{R}^m_+$. Again, as $f_{j,\alpha}$ are convex underestimators of f_j on X^* for all $j \in \{1,\dots,m\}$, we conclude that $p \in \{f_\alpha(x^*)\} + \mathbb{R}^m_+$, which is a contradiction to (4). Thus, X^* contains no efficient point. □

Condition (4) can be tested by solving a convex single objective optimization problem for every local upper bound. Instead of solving such an optimization problem for each point $\bar{p} \in \mathcal{L}_{LUB}$ we solve it for a few points and immediately obtain the information on how to generate and improve an outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$. This information can then be used to efficiently reduce the number of points for which the single objective optimization problem has to be solved. This corresponds to generating tighter lower bounds for the values of f over a box. Thus, let $\bar{p} \in \mathcal{L}_{LUB}$ and let a box X^* be given. We check if $\bar{p} \in f_\alpha(X^*) + \mathbb{R}^m_+$ holds by solving the following convex single objective optimization problem:

$$(P_{\bar{p},X^*}) \qquad \min_{(x,t)\in\mathbb{R}^{n+1}} t \quad \text{s.t. } x \in X^*,\ \bar{p} + te \geq f_\alpha(x).$$

A minimal solution of $(P_{\bar{p},X^*})$ is named $(\tilde{x},\tilde{t})$. If $\tilde{t} \leq 0$, then it holds that $\bar{p} \in f_\alpha(X^*) + \mathbb{R}^m_+$. Otherwise, if $\tilde{t} > 0$, the point $\bar{p}$ lies outside of $f_\alpha(X^*) + \mathbb{R}^m_+$ and can be separated from $f_\alpha(X^*) + \mathbb{R}^m_+$ with a supporting hyperplane. This is used for our new discarding test. The test is applied to a box X^* and consists of a finite number of iterations where an outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ is determined. The initial outer approximation is $\{a\} + \mathbb{R}^m_+$, where a is the ideal point of f_α on X^* (see (2)). In each iteration a local upper bound $\bar{p}$ is chosen. The first step is the comparison of the current outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ with $\bar{p}$ by checking if the inequalities which describe the outer approximation are satisfied. In the case in which $\bar{p}$ is not an element of this outer approximation, the next iteration, namely choosing a next local upper bound, starts. Otherwise we continue with solving $(P_{\bar{p},X^*})$ to obtain the position of $\bar{p}$ w.r.t. $f_\alpha(X^*) + \mathbb{R}^m_+$. The abovementioned cases ($\tilde{t} > 0$, $\tilde{t} \leq 0$) can occur. We extend this to three cases to reduce the effort as only ε-efficiency is the aim for a given scalar $\varepsilon > 0$. Hence, we differentiate between the following cases for the minimum value $\tilde{t}$ of $(P_{\bar{p},X^*})$.

1. $\tilde{t} \leq 0$, i.e., $\bar{p} \in f_\alpha(X^*) + \mathbb{R}^m_+$: Efficient points in X^* are possible. Thus, X^* cannot be discarded and we distinguish between the following two subcases.
   (a) $\tilde{t} < -\frac{\varepsilon}{2}$: Stop the whole discarding test in order to bisect X^* later.
   (b) $-\frac{\varepsilon}{2} \leq \tilde{t} \leq 0$: Construct a supporting hyperplane to improve the outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ and choose the next local upper bound. Moreover, set a flag that X^* cannot be discarded.
2. $\tilde{t} > 0$, i.e., $\bar{p} \notin f_\alpha(X^*) + \mathbb{R}^m_+$: Construct a supporting hyperplane to improve the outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ and choose the next local upper bound.

Only if case 2 holds for every local upper bound can the box X^* be discarded (see Lemma 3.5). The 1(b) case is motivated by obtaining ε-efficient points at the end of the algorithm, as we will prove later. To store X^* in the solution list, there has to be at least one $\bar{p}$ where the 1(b) case is fulfilled and no $\bar{p}$ for which 1(a) holds.

During the discarding test new supporting hyperplanes are constructed if $\tilde{t} \geq -\frac{\varepsilon}{2}$. The support vector of such a hyperplane is $\tilde{y} := \bar{p} + \tilde{t}e$. For calculating a normal vector $\lambda^* \in \mathbb{R}^m$ of the supporting hyperplane, a procedure is given in [13], where a linear single objective optimization problem has to be solved. Alternatively to this and with help of properties of duality theory, we can also use a Lagrange multiplier $\lambda^* \in \mathbb{R}^m$ of the constraint $\bar{p} + te \geq f_\alpha(x)$ to obtain a normal vector of the supporting hyperplane, as explained in detail in [31]. The following algorithm describes the procedure, where the flag $\mathcal{D}$ stands for the decision to discard a box after the algorithm and the flag $\mathcal{B}$ for bisecting the box, respectively.

Algorithm 2. Discarding test.

INPUT: $X^* \in \mathbb{IR}^n,\ f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^m),\ \mathcal{L}_{PNS}, \mathcal{L}_{LUB} \subseteq \mathbb{R}^m,\ \varepsilon > 0,\ \alpha.$
OUTPUT: Flags $\mathcal{D}$, $\mathcal{B}$; lists $\mathcal{L}_{PNS}$, $\mathcal{L}_{LUB}$.

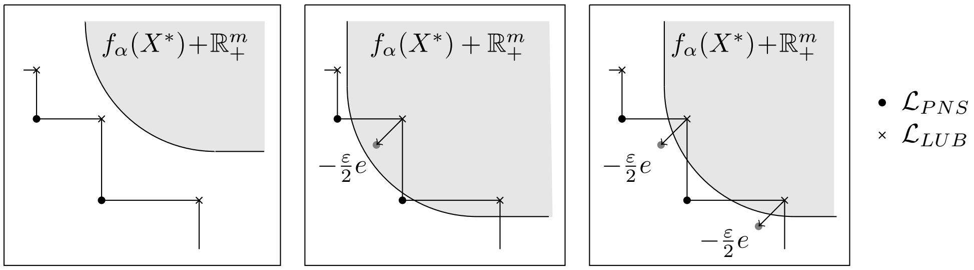
```
1:  Compute for every objective function its convex underestimator on $X^*$ and its
    corresponding minimum $x^j$, update $\mathcal{L}_{PNS}$ by $f(x^j)$ and $\mathcal{L}_{LUB} =: \{p^1, \dots, p^k\}$;
2:  $\mathcal{D} \leftarrow 1,\ \mathcal{B} \leftarrow 0$;
3:  for $s = 1, \dots, k$ do
4:      if $p^s$ is inside the current outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ then
5:          solve $(P_{p^s, X^*})$ with minimal solution $(\tilde{x}, \tilde{t})$;
6:          if $\tilde{t} < -\frac{\varepsilon}{2}$ then
7:              break for-loop
8:              set flags $\mathcal{D} \leftarrow 0$ and $\mathcal{B} \leftarrow 1$;
9:          else if $\tilde{t} \leq 0$ then
10:             update outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ and set flag $\mathcal{D} \leftarrow 0$;
11:         else update outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$.
```

We could actually add in line 1 the image of any other point of X^0, for example, the midpoint of the considered box X^*, to the list $\mathcal{L}_{PNS}$, because we only have box constraints. As we plan to work with general convex constraints later, the feasibility of the preimages of all points of $\mathcal{L}_{PNS}$ has to be ensured. Thus, we need that $\mathcal{L}_{PNS} \subseteq f(M) \subseteq f(X^0)$, which is the case for our construction in line 1. For more information about the handling of constraints, we refer the reader to section 5. Note that the condition of line 4 can be checked by evaluating a finite number of inequalities which are given by the current outer approximation. The following theorem gives the correctness of this discarding test.

THEOREM 3.6. *Let a box $X^* \subseteq X^0 \in \mathbb{IR}^n$ and* (P) *be given. Let $\mathcal{L}_{LUB} \subseteq \mathbb{R}^m$ be a local upper bound set w.r.t. $\mathcal{L}_{PNS}$. If X^* contains an efficient point of* (P)*, then the output of Algorithm* 2 *is $\mathcal{D} = 0$, i.e., X^* will not be discarded by Algorithm* 2.

Proof. Assume that there is some efficient point x^* of (P) with $x^* \in X^*$. Suppose the output of Algorithm 2 applied to X^* is $\mathcal{D} = 1$. This means that for all local upper bounds either the conditions in lines 6 and 9 are not satisfied or they are not contained in the current outer approximation (see line 4). In the case in which the latter occurs the local upper bound is clearly not in $f_\alpha(X^*)+\mathbb{R}^m_+$. For the other local upper bounds $\bar{p} \in \mathcal{L}_{LUB}$, which do not satisfy lines 6 and 9, but do satisfy line 4, the optimization problem $(P_{\bar{p},X^*})$ has a minimal value $\tilde{t} > 0$. Hence, it holds for all $\bar{p} \in \mathcal{L}_{LUB}$ that $\bar{p} \notin f_\alpha(X^*) + \mathbb{R}^m_+$ and with Lemma 3.5 we have a contradiction to the assumption that X^* contains an efficient point. □

All possibilities for the results of the discarding test applied to a box X^* in the two-dimensional case are illustrated in Figures 5(a) to 5(c).


$f_\alpha(X^*)+\mathbb{R}^m_+$
$f_\alpha(X^*)+\mathbb{R}^m_+$
$f_\alpha(X^*)+\mathbb{R}^m_+$
$-\frac{\varepsilon}{2}e$
$-\frac{\varepsilon}{2}e$
$-\frac{\varepsilon}{2}e$
• $\mathcal{L}_{PNS}$
× $\mathcal{L}_{LUB}$

(a) *Discard box X^*.* (b) *Do not discard box X^*, but bisect it.* (c) *Neither discard box X^* nor bisect it.*

FIG. 5. *Possible situations during a discarding test applied to box X^*.*

The list $\mathcal{L}_{PNS}$ is growing during the algorithm. Hence, we obtain better upper bounds for the nondominated set. It is possible that a box X^* was not discarded and not bisected by Algorithm 2, while it could be discarded if we compared it with the final list $\mathcal{L}_{PNS}$. Thus, the whole algorithm, which is presented later, consists of more than one `while`-loop in the scheme of Algorithm 1. The first loop executes the discarding test presented in Algorithm 2. In the second loop the lists $\mathcal{L}_{PNS}$ and $\mathcal{L}_{LUB}$ are static and will not be updated as in line 1 of Algorithm 2. Nevertheless, the discarding test is applied in the same way as before. This method can be found in Algorithm 3. Additionally, in that procedure we reuse the already calculated approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ and, what is more, we save the feasible points $\tilde{x}$ which are a minimal solution of $(P_{\bar{p},X^*})$ if its corresponding $\tilde{t}$ is between $-\frac{\varepsilon}{2}$ and 0. Note that the $\tilde{t} < -\frac{\varepsilon}{2}$ case is not possible for a box X^* which passed Algorithm 2 with any bisecting ($\mathcal{B} = 1$). This fact will be shown in Lemma 4.5. The points $\tilde{x}$ with $-\frac{\varepsilon}{2} \leq \tilde{t} \leq 0$ are collected in the list $\mathcal{X}$ and will serve as the possible points of the (ε,δ)-efficient set $\mathcal{A}$.

THEOREM 3.7. *Let a box $X^* \subseteq X^0 \in \mathbb{IR}^n$ and* (P) *be given. Let $\mathcal{L}_{LUB} \subseteq \mathbb{R}^m$ be a local upper bound set w.r.t. $\mathcal{L}_{PNS}$. If X^* contains an efficient point of* (P)*, then the output of Algorithm* 2 *is $\mathcal{D} = 0$, i.e., X^* will not be discarded by Algorithm* 3.

Algorithm 3. Discarding test with static lists $\mathcal{L}_{PNS}$, $\mathcal{L}_{LUB}$.

```
INPUT: $X^* \in \mathbb{IR}^n,\ f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^m),\ \mathcal{L}_{PNS},\ \mathcal{L}_{LUB} = \{p^1, \dots, p^k\} \subseteq \mathbb{R}^m,\ \varepsilon > 0,\ \alpha$.
OUTPUT: List $\mathcal{X}$, flag $\mathcal{D}$.
if there is no current outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ calculated yet then
    calculate ideal point $a$ of $f_\alpha$ and initialize $\{a\} + \mathbb{R}^m_+$ as an outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$;
$\mathcal{D} \leftarrow 1,\ \mathcal{X} \leftarrow \emptyset$;
for $s = 1, \dots, k$ do
    if $p^s$ is inside the current outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$ then
        solve $(P_{p^s, X^*})$ with minimal solution $(\tilde{x}, \tilde{t})$;
        update outer approximation of $f_\alpha(X^*) + \mathbb{R}^m_+$;
        if $\tilde{t} \le 0$ then  $\mathcal{X} \leftarrow \mathcal{X} \cup \tilde{x}$ and set flag $\mathcal{D} \leftarrow 0$.
```

Proof. The proof is analogous to the proof of Theorem 3.6. □

3.3. The complete algorithm.

Having now the new discarding test, we can present the complete algorithm together with the new termination procedure. The whole algorithm for (P) is given in Algorithm 4.

The algorithm consists of three `while`-loops. The list $\mathcal{L}_{S,t},\ t = 1, 2, 3$, is the solution list of the tth `while`-loop and becomes the working list for the next loop if $t = 1, 2$. The first loop from line 3 handles the basic discarding test, which was explained on page 803 in detail. It generates the list $\mathcal{L}_{PNS}$ until this list is close to the nondominated set in dependence of ε. All boxes X from $\mathcal{L}_{S,1}$ have the following properties:

$$\exists \bar{p} \in \mathcal{L}_{LUB} : \bar{p} \in f_\alpha(X) + \mathbb{R}^m_+, \tag{5}$$

$$\forall p \in \mathcal{L}_{LUB} : p - \frac{\varepsilon}{2} e \notin f_\alpha(X) + \mathbb{R}^m_+. \tag{6}$$

The first property is true, because if all local upper bounds were outside of $f_\alpha(X) + \mathbb{R}^m_+$, then the box X would be discarded (see Lemma 3.5 for the proof). If the second property did not hold for X, this box would not be stored in the solution list $\mathcal{L}_{S,1}$, because line 6 of Algorithm 2 would be satisfied and X would be bisected into subboxes.

With the second and third `while`-loop of Algorithm 4 we do not lose characteristics (5) and (6). The second loop only checks whether some boxes from $\mathcal{L}_{S,1}$ can be discarded by the final lists $\mathcal{L}_{PNS}$ and $\mathcal{L}_{LUB}$. In the third loop we check for every box $X \in \mathcal{L}_{S,2}$ whether $\omega(X)$ is less than or equal to δ for a predefined $\delta > 0$. If the box X is not small enough, we bisect it and apply the discarding test to both subboxes. Moreover, we compute the (ε, δ)-efficient set $\mathcal{A}$, which can be stated as

$$\begin{aligned}\mathcal{A} := {} & \{x \in X^0 \mid f(x) \in \mathcal{L}_{PNS}\} \\ & \cup \bigcup_{X^* \in \mathcal{L}_{S,2}} \left\{ x \in X^* \;\middle|\; \begin{array}{l} (\exists\, \bar{p} \in \mathcal{L}_{LUB}, t \le 0 : (x, t) \text{ is a min. solution of } (P_{\bar{p}, X^*})) \\ \wedge\, (\omega(X^*) \le \delta) \\ \wedge\, \left((\exists \tilde{p} \in \mathcal{L}_{LUB} : f(x) \le \tilde{p}) \vee (\omega(X^*) < \sqrt{\tfrac{\varepsilon}{\alpha}})\right) \end{array} \right\}.\end{aligned}$$

Note that the union is not a disjoint union. The proof of the (ε, δ)-efficiency of $\mathcal{A}$ is a main part of subsection 4.2. To explain the definition of the sets in the second line of the definition of $\mathcal{A}$ we have to take a closer look at the third `while`-loop: When a subbox X^* is considered in this loop, $(P_{\bar{p}, X^*})$ is solved for some local upper

Algorithm 4. Algorithm to find an (ε, δ)-efficient set of (P).

```
INPUT: $X^0 \in \mathbb{IR}^n,\ f \in \mathcal{C}^2(\mathbb{R}^n, \mathbb{R}^m),\ \varepsilon > 0,\ \delta > 0$.
OUTPUT: $\mathcal{A},\ \mathcal{L}_{S,3},\ \mathcal{L}_{PNS},\ \mathcal{L}_{LUB}$.
 1: $\mathcal{L}_W \leftarrow \{X^0\},\ \mathcal{L}_{S,1} \leftarrow \emptyset,\ \mathcal{L}_{S,2} \leftarrow \emptyset,\ \mathcal{L}_{S,3} \leftarrow \emptyset,\ \mathcal{A} \leftarrow \emptyset,\ \mathcal{L}_{PNS} \leftarrow \emptyset,\ \mathcal{L}_{LUB} \leftarrow \emptyset$;
 2: calculate $\alpha$ such that $f_{j,\alpha}$ is a convex underestimator of $f_j$ on $X^0,\ j = 1, \dots, m$;
 3: while $\mathcal{L}_W \neq \emptyset$ do
 4:     select a box $X^*$ from $\mathcal{L}_W$ and delete it from $\mathcal{L}_W$;
 5:     bisect $X^*$ perpendicularly to a direction of maximum width $\rightarrow X^1, X^2$;
 6:     for $l = 1, 2$ do
 7:         apply Algorithm 2 to $X^l$;
 8:         if $\mathcal{B} = 1$ then store $X^l$ in $\mathcal{L}_W$;
 9:         else if $\mathcal{D} = 0$ then store $X^l$ in $\mathcal{L}_{S,1}$;
10:         else discard $X^l$;
11: while $\mathcal{L}_{S,1} \neq \emptyset$ do
12:     select a box $X^*$ from $\mathcal{L}_{S,1}$ and delete it from $\mathcal{L}_{S,1}$;
13:     apply Algorithm 3 to $X^*$;
14:     if $\mathcal{D} = 0$ then store $X^*$ in $\mathcal{L}_{S,2}$;
15:     else discard $X^*$;
16: while $\mathcal{L}_{S,2} \neq \emptyset$ do
17:     select a box $X^*$ from $\mathcal{L}_{S,2}$ and delete it from $\mathcal{L}_{S,2}$;
18:     apply Algorithm 3 to $X^*$ and obtain $\mathcal{X}$;
19:     if $\mathcal{D} = 1$ then discard $X^*$;
20:     else if $\mathcal{D} = 0$ and $\omega(X^*) \leq \delta$ then
21:         $\{x^1, \dots, x^k\} \leftarrow \mathcal{X}$;
22:         for $s = 1, \dots, k$ do
23:             if $f(x^s) \leq \bar{p}$ for at least one $\bar{p} \in \mathcal{L}_{LUB}$ or $\omega(X^*) < \sqrt{\frac{\varepsilon}{\alpha}}$ then
24:                 $\mathcal{A} \leftarrow \mathcal{A} \cup \{x^s\}$;
25:                 store $X^*$ in $\mathcal{L}_{S,3}$;
26:     if $\mathcal{D} = 0$ and no point of $\mathcal{X}$ was stored in $\mathcal{A}$ then
27:         bisect $X^*$ perpendicularly to a direction of maximum width $\rightarrow X^1, X^2$;
28:         store $X^1$ and $X^2$ in $\mathcal{L}_{S,2}$;
29: $\mathcal{A} \leftarrow \mathcal{A} \cup \{x \in X^0 \mid f(x) \in \mathcal{L}_{PNS}\}$.
```

bounds $\bar{p}$. A point x, which belongs to a minimal solution (x, t) of $(P_{\bar{p}, X^*})$ and with a nonpositive t, is a possible candidate for an ε-efficient point and thus $x \in \mathcal{X}$. If $f(x)$ is less than or equal to a local upper bound, then x is added to $\mathcal{A}$. Furthermore, if $\omega(X)$ is bounded by $\sqrt{\frac{\varepsilon}{\alpha}}$ (usually smaller than δ) we add all elements of $\mathcal{X}$ to $\mathcal{A}$. Note that $\mathcal{X} \neq \emptyset$ if and only if $\mathcal{D} = 0$. Moreover, the preimages of the points of the final list $\mathcal{L}_{PNS}$, i.e., points of the first set in the description of $\mathcal{A}$, are also added to $\mathcal{A}$ at the end of the algorithm.

4. Convergence results.

In this section we show the correctness and finiteness of Algorithm 4.

4.1. Termination.

To show the termination of the algorithm we have to verify that each `while`-loop of Algorithm 4 is finite. We start by showing the termination of the first `while`-loop.

LEMMA 4.1. *The first* `while`*-loop (lines* 3–10*) of Algorithm* 4 *terminates.*

Proof. Assume the first `while`-loop does not terminate. Hence, there must be an infinite sequence of boxes $X^0 \supset X^1 \supset \cdots \supset X^k \supset \cdots$ which were not discarded, but bisected after applying Algorithm 2 on each box. Thus, every box X^k will be stored in $\mathcal{L}_W$ and bisected in another iteration, where the box X^{k+1} is one of the two obtained subboxes. Obviously, the box width decreases among the sequence of boxes, i.e., $\omega(X^k) \geq \omega(X^{k+1})$ for every $k \in \mathbb{N}$ and converges to 0 (because we divide the boxes perpendicular to a side with maximal width). For δ_{X^0} as in (1) choose the first box $X^{\tilde{k}}$ with $\omega(X^{\tilde{k}}) \leq \delta_{X^0}$. Let $a \in \mathbb{R}^m$ be the ideal point of f on $X^{\tilde{k}}$ (see (2)). From Lemma 2.3 for every objective function it follows that for all $x \in X^{\tilde{k}}$, $|f_j(x) - a_j| \leq \frac{\varepsilon}{2}$ for all $j = 1, \dots, m$ or

$$f(x) \in (\{a\} + \mathbb{R}^m_+) \cap \left(\left\{ a + \frac{\varepsilon}{2} e \right\} - \mathbb{R}^m_+ \right). \tag{7}$$

Consider now the minima of each convex underestimator. Choose an arbitrary $\tilde{x}^j \in \operatorname{argmin}\{f_{j,\alpha}(x) \mid x \in X^{\tilde{k}}\}$ for every $j = 1, \dots, m$. The images of the points $\tilde{x}_j$ under the original function, i.e., $f(\tilde{x}^j)$, are potential points of the list $\mathcal{L}_{PNS}$ (see line 1 of Algorithm 2), and clearly satisfy (7) as well. Let us choose one of these and denote it by q. This point will be added to $\mathcal{L}_{PNS}$ if there is no other point from the current list $\mathcal{L}_{PNS}$ dominating q.

Because of the assumption that $X^{\tilde{k}}$ will be bisected, there must be a local upper bound $\bar{p}$ with the minimal solution $(\tilde{x}, \tilde{t})$ of $(P_{\bar{p}, X^{\tilde{k}}})$, where $\tilde{t}$ is less than $-\varepsilon/2$. Now, we want to check for each local upper bound if this is possible. If we show that for any local upper bound $\bar{p}$ we have that the minimal solution of $(P_{\bar{p}, X^{\tilde{k}}})$ is never less than $-\varepsilon/2$, we can thus conclude that the assumption is wrong and $X^{\tilde{k}}$ will not be bisected.

First, consider all local upper bounds which do not belong to $\{a\} + \mathbb{R}^m_+$. Therefore, let $\bar{p} \in \mathcal{L}_{LUB} \setminus (\{a\} + \mathbb{R}^m_+)$, i.e., there is a $u \in \{1, \dots, m\}$ with $\bar{p}_u < a_u$. The condition in line 4 of Algorithm 2 is not satisfied and $(P_{\bar{p}, X^{\tilde{k}}})$ will not be solved.

Next, we consider those $\bar{p} \in \mathcal{L}_{LUB}$ with $\bar{p} \in \mathcal{L}_{LUB} \cap (\{a\} + \mathbb{R}^m_+)$, in case there are any, and distinguish two cases, the first of which is

$$|\bar{p}_u - a_u| = \bar{p}_u - a_u \leq \frac{\varepsilon}{2} \text{ for one } u \in \{1, \dots, m\}. \tag{8}$$

Problem $(P_{\bar{p}, X^{\tilde{k}}})$ is solved and has a minimal solution $(\tilde{x}, \tilde{t})$. Then it holds that $\bar{p}_u + \tilde{t} \geq f_{u,\alpha}(\tilde{x}) \geq a_u$. Hence, $\frac{\varepsilon}{2} \geq \bar{p}_u - a_u \geq -\tilde{t}$, which leads to $\tilde{t} \geq -\frac{\varepsilon}{2}$. The second case is that there is some $\bar{p} \in \mathcal{L}_{LUB} \cap (\{a\} + \mathbb{R}^m_+)$ with

$$|\bar{p}_j - a_j| = \bar{p}_j - a_j > \frac{\varepsilon}{2} \text{ for all } j \in \{1, \dots, m\} \tag{9}$$

or equivalently $\bar{p} \in \{a + \frac{\varepsilon}{2} e\} + \operatorname{int}(\mathbb{R}^m_+)$. It follows for every $j \in \{1, \dots, m\}$ that $a_j + \frac{\varepsilon}{2} < \bar{p}_j$. But we know there is a point q (see above) which is a candidate for $\mathcal{L}_{PNS}$ and belongs to the set $(\{a\} + \mathbb{R}^m_+) \cap (\{a + \frac{\varepsilon}{2} e\} - \mathbb{R}^m_+)$. Define

$$y := \begin{cases} q' & \text{if there is a } q' \in \mathcal{L}_{PNS} \text{ with } q' \leq q, \\ q & \text{otherwise,} \end{cases}$$

and thus $y \in \mathcal{L}_{PNS}$. Now, we obtain by considering each component of a, y, and $\bar{p}$ the inequalities $y_j \leq q_j \leq a_j + \frac{\varepsilon}{2} < \bar{p}_j$ for all $j = 1, \dots, m$. Hence, y strictly dominates $\bar{p}$, which is a contradiction to Lemma 3.3(i). Thus, the existence of a local upper bound in $\{a+\frac{\varepsilon}{2}e\}+\operatorname{int}(\mathbb{R}^m_+)$ is not possible.

Clearly, it is not possible for $X^{\tilde{k}}$ that it satisfies the conditions for bisection. Hence, the assumed infinite sequence of subboxes does not exist. According to this the first `while`-loop will terminate. □

LEMMA 4.2. *The second* `while`*-loop (lines* 11–15*) of Algorithm* 4 *terminates.*

Proof. The termination of the second `while`-loop is clear, because it has exactly $|\mathcal{L}_{S,1}|$ iterations. □

LEMMA 4.3. *The third* `while`*-loop (lines* 16–28*) of Algorithm* 4 *terminates.*

Proof. Assume the third `while`-loop does not terminate. Hence, there must be an infinite sequence of boxes $X^1 \supset \cdots \supset X^k \supset \cdots$ with $X^1 \in \mathcal{L}_{S,2}$ after the second loop, which were not discarded, but bisected after applying Algorithm 3 on each box. Hence, every box X^k will be stored in $\mathcal{L}_{S,2}$ and bisected in another iteration, where the box X^{k+1} is one of the two obtained subboxes. Obviously, the box width decreases among the sequence of boxes, i.e., $\omega(X^k) \geq \omega(X^{k+1})$ for every $k \in \mathbb{N}$ and converges to 0 (because we divide the boxes perpendicular to a side with maximal width). Let us choose the first box $X^{\tilde{k}}$ with $\omega(X^{\tilde{k}}) < \min\{\delta, \sqrt{\frac{\varepsilon}{\alpha}}\}$. For $X^{\tilde{k}}$ we have $\mathcal{D}=0$, otherwise it will be discarded. Therefore, the conditions in lines 20 and 23 of Algorithm 4 are satisfied and $X^{\tilde{k}}$ will be stored in $\mathcal{L}_{S,3}$. This contradicts the assumption that $X^{\tilde{k}}$ will be bisected. Hence, the assumed infinite sequence of subboxes does not exist. According to this the third `while`-loop will terminate. □

With these lemmas we obtain that the whole algorithm terminates.

4.2. Correctness. First, we state that all efficient points x of (P) are contained in the union of boxes from the final list $\mathcal{L}_{S,3}$.

LEMMA 4.4. *Let $\mathcal{L}_{S,3}$ be the output of Algorithm* 4 *for arbitrary $\varepsilon,\delta>0$ and let $\mathcal{L}_{S,1}$ and $\mathcal{L}_{S,2}$ be the lists after the first and the second* `while`*-loops, respectively. Then*

$$X_E \subseteq \bigcup_{X\in\mathcal{L}_{S,3}} X \subseteq \bigcup_{X\in\mathcal{L}_{S,2}} X \subseteq \bigcup_{X\in\mathcal{L}_{S,1}} X.$$

Proof. This is a direct consequence of Theorems 3.6 and 3.7 and the way the lists are constructed. □

The next two lemmas show the fact, which was mentioned in subsection 3.2 on page 805, that in Algorithm 3 the case in which $\tilde{t} < -\frac{\varepsilon}{2}$ is not possible in the second and third `while`-loops of Algorithm 4.

LEMMA 4.5. *Let $X \in \mathbb{IR}^m$ be chosen from the working list $\mathcal{L}_{S,1}$ during the second* `while`*-loop of Algorithm* 4 *and hence be an input for Algorithm* 3*. If $(P_{\bar{p},X})$ is solved for any $\bar{p} \in \mathcal{L}_{LUB}$ within Algorithm* 3*, we obtain a minimal solution $(\tilde{x},\tilde{t})$ with $\tilde{t} \geq -\frac{\varepsilon}{2}$.*

Proof. Let $\bar{p} \in \mathcal{L}_{LUB}$ be inside the current outer approximation of $f_\alpha(X)+\mathbb{R}^m_+$ and let $(\tilde{x},\tilde{t})$ be the minimal solution of $(P_{\bar{p},X})$. Assume now that $\tilde{t} < -\frac{\varepsilon}{2}$. In particular, by $f_\alpha(\tilde{x}) \leq \bar{p}+\tilde{t}e < \bar{p}$ we obtain $\bar{p} \in f_\alpha(X)+\mathbb{R}^m_+$. Because of $X \in \mathcal{L}_{S,1}$ this box was not discarded in the first `while`-loop of Algorithm 4, i.e., $\mathcal{D}=0$ and $\mathcal{B}=0$. For the next steps we consider X during the first `while`-loop, where Algorithm 2 is executed. Let $\mathcal{L}'_{LUB}$ be the set of local upper bounds at this time. Then it holds that

$$(10)\qquad \forall p' \in \mathcal{L}'_{LUB}:\ (P_{p',X}) \text{ was solved with minimal solution } (x',t') \Rightarrow t' \geq -\frac{\varepsilon}{2}.$$

Now, we distinguish two cases, the first of which is $\bar{p} \in \mathcal{L}'_{LUB}$. Because $\bar{p} \in f_\alpha(X) + \mathbb{R}^m_+$, problem $(P_{\bar{p},X})$ was solved. As $(\tilde{x},\tilde{t})$ is feasible for $(P_{\bar{p},X})$ with $\tilde{t} < -\frac{\varepsilon}{2}$, this contradicts (10).

The second case is $\bar{p} \notin \mathcal{L}'_{LUB}$, i.e., $\bar{p}$ was added to the set of local upper bounds after X was considered in the first `while`-loop. Then there exists a $p^* \in \mathcal{L}'_{LUB}$ with $\bar{p} \leq p^*$ and $\bar{p} \neq p^*$. This fact can easily be seen by induction over a generating algorithm of a local upper bound set, for example, Algorithm 3 in [27]. Because of $\bar{p} \in f_\alpha(X) + \mathbb{R}^m_+$ it also holds that $p^* \in f_\alpha(X) + \mathbb{R}^m_+$ and the optimization problem $(P_{p^*,X})$ was solved in Algorithm 2. Then $(\tilde{x},\tilde{t})$ is also feasible for $(P_{p^*,X})$, because $f_\alpha(\tilde{x}) \leq \bar{p} + \tilde{t}e \leq p^* + \tilde{t}e$, which contradicts (10). □

LEMMA 4.6. *Let $X \in \mathbb{IR}^m$ be chosen from the working list in $\mathcal{L}_{S,2}$ during the third `while`-loop of Algorithm* 4 *and hence be an input for Algorithm* 3. *If $(P_{\bar{p},X})$ is solved for any $\bar{p} \in \mathcal{L}_{LUB}$ within Algorithm* 3, *we obtain a minimal solution $(\tilde{x},\tilde{t})$ with $\tilde{t} \geq -\frac{\varepsilon}{2}$.*

Proof. Let $\bar{p} \in \mathcal{L}_{LUB}$ be inside the current outer approximation of $f_\alpha(X) + \mathbb{R}^m_+$ and let $(\tilde{x},\tilde{t})$ be the minimal solution of $(P_{\bar{p},X})$. Assume now that $\tilde{t} < -\frac{\varepsilon}{2}$. In particular, by $f_\alpha(\tilde{x}) \leq \bar{p} + \tilde{t}e < \bar{p}$ we obtain $\bar{p} \in f_\alpha(X) + \mathbb{R}^m_+$. Note that the set $\mathcal{L}_{LUB}$ is fixed after the first loop.

Because of Lemma 4.5 the box X was not considered in the second `while`-loop, i.e., $X \notin \mathcal{L}_{S,2}$ after line 15. But there exists a box X^* with $X \subseteq X^*$ which has been considered and not discarded in this loop. Let f^*_α be the componentwise convex underestimator of f on $X^* = [\underline{x^*}, \overline{x^*}]$, i.e., $f^*_{j,\alpha}(x) = f(x) - \frac{\alpha}{2}(\underline{x^*} - x)^T(\overline{x^*} - x)$ for all $x \in X^*$, $j = 1,\ldots,m$, which is clearly less than f_α on X, i.e., $f^*_\alpha(x) \leq f_\alpha(x)$ for all $x \in X$. Because of $\bar{p} \in f_\alpha(X) + \mathbb{R}^m_+$ it also holds that $\bar{p} \in f^*_\alpha(X^*) + \mathbb{R}^m_+$ and the optimization problem $(P_{\bar{p},X^*})$ was solved in Algorithm 3. Let (x^*,t^*) be a minimal solution of $(P_{\bar{p},X^*})$. Because of Lemma 4.5 we have $t^* \geq -\frac{\varepsilon}{2}$. Since $X \subseteq X^*$ and $f_\alpha(\tilde{x}) \geq f^*_\alpha(\tilde{x})$ the pair $(\tilde{x},\tilde{t})$ is also feasible for $(P_{\bar{p},X^*})$, which contradicts the minimality of (x^*,t^*). □

Based on the list $\mathcal{L}_{LUB}$ and thus on the list $\mathcal{L}_{PNS}$, which are calculated by Algorithm 4, one obtains a set which contains all nondominated points of (P). We prove this in Theorem 4.8. This set, which looks like a staircase-shaped tube or pipe in the two-dimensional case, is defined as follows:

$$T \;:=\; \left(\bigcup\nolimits_{\bar{p}\in\mathcal{L}_{LUB}} \{\bar{p}\} - \mathbb{R}^m_+\right) \setminus \left(\bigcup\nolimits_{\bar{p}\in\mathcal{L}_{LUB}} \left\{\bar{p} - \tfrac{\varepsilon}{2}e\right\} - \operatorname{int}(\mathbb{R}^m_+)\right). \tag{11}$$

An illustration of such a set T is shown in Figure 6. Recall that $\hat{Z}$ was defined as a box with $f(X^0) \subseteq \operatorname{int}(\hat{Z})$.

The set T contains all points from $\mathcal{L}_{PNS}$, which will be shown in the next lemma.

LEMMA 4.7. *Let $\mathcal{L}_{LUB}$ be the local upper bound set w.r.t. $\mathcal{L}_{PNS}$. Let T be defined as in* (11). *Then $\mathcal{L}_{PNS} \subseteq T$.*

Proof. Because of Lemma 3.4 there exists for every point $q \in \mathcal{L}_{PNS}$ a point $p^q \in \mathcal{L}_{LUB}$ with $q \in \{p^q\} - \mathbb{R}^m_+$. Assume q belongs to a set $\{p - \frac{\varepsilon}{2}e\} - \operatorname{int}(\mathbb{R}^m_+)$ for any $p \in \mathcal{L}_{LUB}$. Then it holds for every $j \in \{1,\ldots,m\}$ that $q_j < p_j - \frac{\varepsilon}{2} < p_j$. Certainly, this contradicts Lemma 3.3(i). Thus, $\mathcal{L}_{PNS} \subseteq T$. □

THEOREM 4.8. *Let $\mathcal{L}_{LUB}$ be the local upper bound set w.r.t. $\mathcal{L}_{PNS}$ with $\mathcal{L}_{PNS}$ from Algorithm* 4, *and let T be defined as in* (11). *Let $\bar{x}$ be an efficient point of* (P). *Then $f(\bar{x}) \in T$, i.e., the nondominated set of* (P) *is a subset of T.*

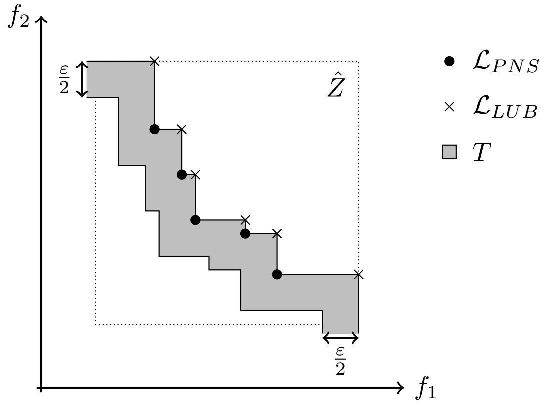


FIG. 6. *The set T, which depends on $\frac{\varepsilon}{2}$; $m = 2$.*

Proof. Let $\bar{x} \in X_E$, i.e., $\bar{x}$ is an efficient point of (P). We assume that $f(\bar{x}) \notin T$. There are two possibilities: First, $f(\bar{x})$ lies above T, i.e., $f(\bar{x}) \in \hat{Z} \setminus (T - \mathbb{R}^m_+)$. By $f(\bar{x}) \notin T - \mathbb{R}^m_+$, it holds for all $p \in \mathcal{L}_{LUB}$ that $f(\bar{x}) \nleq p$. In particular, $f(\bar{x}) \notin S$, otherwise we have a contradiction to Definition 3.2(i). Using the definition of S (see (3)) it follows that there exists a point $q \in \mathcal{L}_{PNS}$ with $q \leq f(\bar{x})$. Since q is an image of a feasible point and $\bar{x}$ is efficient, $q = f(\bar{x})$ holds. However, we have assumed that $f(\bar{x}) \notin T$, which contradicts $q \in \mathcal{L}_{PNS} \subseteq T$. The other case is $f(\bar{x}) \in (T - \mathbb{R}^m_+) \setminus T$. Hence, there is a local upper bound $\bar{p} \in \mathcal{L}_{LUB}$ with $f(\bar{x}) < \bar{p} - \frac{\varepsilon}{2}e$, which leads to the chain of inequalities $f_\alpha(\bar{x}) \leq f(\bar{x}) < \bar{p} - \frac{\varepsilon}{2}e$. By Lemma 4.4 there is some box $X \in \mathcal{L}_{S,1}$ with $\bar{x} \in X$ and thus $\bar{p} - \frac{\varepsilon}{2} \in f_\alpha(X) + \mathbb{R}^m_+$. This is a contradiction to (6). Therefore, $f(\bar{x})$ belongs to T. □

Remark 4.9. As a consequence of Theorem 4.8 we also have $f(X^0) \subseteq T + \mathbb{R}^m_+$, i.e., $f(X^0) \cap (\mathbb{R}^m \setminus (T + \mathbb{R}^m_+)) = \emptyset$, which means that no image of a feasible point of (P) lies below T. This is due to the fact that the ordering cone $\mathbb{R}^m_+$ is a pointed closed convex cone and $f(X^0)$ is a compact set, and thus external stability holds (cf. [39, Theorem 3.2.9]).

Next, we want to show the (ε, δ)-efficiency of the output set $\mathcal{A}$. Recall that the set $\mathcal{X}$ is calculated individually for every considered subbox X^* in line 8 of Algorithm 3. All $x \in \mathcal{X}$ may be ε-efficient points of (P). In Algorithm 4 line 23 two conditions are checked and we will prove in the next lemmas that for those $x \in \mathcal{X}$ which satisfy one of the conditions in line 23, ε-efficiency indeed holds.

LEMMA 4.10. *Every subbox $X \in \mathbb{IR}^n$ of X^0 which is not discarded in the third* `while`*-loop of Algorithm* 4 *and with $\omega(X) < \sqrt{\frac{\varepsilon}{\alpha}}$ contains a point $\tilde{x}$ which is ε-efficient.*

Proof. Set $\tilde{\delta} := \sqrt{\frac{\varepsilon}{\alpha}}$ and let $X \subseteq X^0$ be a box with $\omega(X) < \tilde{\delta}$. By Remark 2.2 we know that for arbitrary $x \in X$ and all $j = 1, \ldots, m$ it holds that $f_j(x) - f_{j,\alpha}(x) = |f_j(x) - f_{j,\alpha}(x)| \leq \frac{\alpha}{2}\omega(X)^2 < \frac{\alpha}{2} \cdot \frac{\varepsilon}{\alpha} = \frac{\varepsilon}{2}$. As f_α is a convex underestimator of f (componentwise) we obtain

$$f_{j,\alpha}(x) \leq f_j(x) < f_{j,\alpha}(x) + \frac{\varepsilon}{2} \text{ for every } j \in \{1, \ldots, m\} \text{ and all } x \in X, \tag{12}$$

which is equivalent to $f(x) \in (\{f_\alpha(x) + \frac{\varepsilon}{2}e\} - \operatorname{int}(\mathbb{R}^m_+)) \cap (\{f_\alpha(x)\} + \mathbb{R}^m_+)$. The box X

was not discarded in the third while-loop. Hence, Algorithm 3 applied to X delivered the output $\mathcal{D} = 0$ and a list $\mathcal{X} \neq \emptyset$; see line 8 of Algorithm 3. Consider an $\tilde{x} \in X$ which was calculated in line 6 of Algorithm 3 w.r.t. the corresponding local upper bound $p^s \in \mathcal{L}_{LUB}$. Note that for the minimal solution $(\tilde{x}, \tilde{t})$ of $(P_{p^s,X})$ we have $-\frac{\varepsilon}{2} \leq \tilde{t} \leq 0$. Recall that $-\frac{\varepsilon}{2} \leq \tilde{t}$ holds because of Lemma 4.6. Suppose now that there is an $\hat{x} \in X^0$ with $f(\hat{x}) \leq f(\tilde{x}) - \varepsilon e$ and $f(\hat{x}) \neq f(\tilde{x}) - \varepsilon e$, which is equivalent to $f(\hat{x}) \in \{f(\tilde{x})\} - \{\varepsilon e\} - (\mathbb{R}^m_+ \setminus \{0_m\})$. Hence, with the above shown property (12) we conclude that

$$\begin{aligned} f(\hat{x}) &\in \{f(\tilde{x})\} - \{\varepsilon e\} - (\mathbb{R}^m_+ \setminus \{0_m\}) \\ &\subseteq \left(\left(\left\{ f_\alpha(\tilde{x}) + \frac{\varepsilon}{2} e \right\} - \operatorname{int}(\mathbb{R}^m_+) \right) \cap \left(\{f_\alpha(\tilde{x})\} + \mathbb{R}^m_+ \right) \right) - \{\varepsilon e\} - (\mathbb{R}^m_+ \setminus \{0_m\}) \\ &\subseteq \left\{ f_\alpha(\tilde{x}) - \frac{\varepsilon}{2} e \right\} - \operatorname{int}(\mathbb{R}^m_+), \end{aligned} \tag{13}$$

which is equivalent to $f(\hat{x}) < f_\alpha(\tilde{x}) - \frac{\varepsilon}{2}$.

As $(\tilde{x}, \tilde{t})$ is a feasible point of $(P_{p^s,X})$ it holds that $p^s + \tilde{t}e \geq f_\alpha(\tilde{x})$. Given this and $\tilde{t} \leq 0$, we obtain $f_\alpha(\tilde{x}) \leq p^s$. By (13) it follows that $f(\hat{x}) < p^s - \frac{\varepsilon}{2}$ and thus $f(\hat{x})$ is not in $T + \mathbb{R}^m_+$, which contradicts Remark 4.9 that no feasible image is below T. Hence, $\tilde{x}$ is ε-efficient. □

LEMMA 4.11. *Let $\mathcal{A}$ be the set generated by Algorithm* 4. *Then $\tilde{x} \in \mathcal{A}$ is an ε-efficient point of* (P).

Proof. Let $\tilde{x}$ be an arbitrary element of $\mathcal{A}$. If $\tilde{x}$ is added in line 29 to $\mathcal{A}$, the point $f(\tilde{x})$ is an element of $\mathcal{L}_{PNS}$ and thus by Lemma 3.4 there is a local upper bound $\bar{p} \in \mathcal{L}_{LUB}$ with $f(\tilde{x}) \leq \bar{p}$. Suppose there is an $\hat{x} \in X^0$ with $f(\hat{x}) \leq f(\tilde{x}) - \varepsilon e$ and $f(\hat{x}) \neq f(\tilde{x}) - \varepsilon e$. With $f(\tilde{x}) \leq \tilde{p}$ we obtain $f(\hat{x}) \leq \tilde{p} - \varepsilon e$ and hence $f(\hat{x}) \notin T$, but $f(\hat{x})$ lies below T. That contradicts Remark 4.9.

If $\tilde{x}$ is added in line 24, $\tilde{x}$ belongs to a box X^* with $\omega(X^*) \leq \delta$ which was not discarded in the third while-loop and, moreover, $\tilde{x} \in \mathcal{X}$. Hence, there is a local upper bound $\bar{p} \in \mathcal{L}_{LUB}$ such that $(\tilde{x}, \tilde{t})$ is a minimal solution of $(P_{\bar{p},X^*})$ and $-\frac{\varepsilon}{2} \leq \tilde{t} \leq 0$. Recall that $-\frac{\varepsilon}{2} \leq \tilde{t}$ holds because of Lemma 4.6. As the first condition in line 23 of Algorithm 4 is satisfied, we have $f(\tilde{x}) \leq \tilde{p}$ for a local upper bound $\tilde{p} \in \mathcal{L}_{LUB}$. By the same arguments as at the beginning of this proof, we can show that $\tilde{x}$ is ε-efficient.

If the first condition in line 23 is not satisfied, but the second one is, i.e., $\omega(X^*) < \sqrt{\varepsilon/\alpha}$, we have to show that each $x \in \mathcal{X}$ is ε-efficient. For this we refer the reader to the proof of Lemma 4.10, because the points in $\mathcal{X}$ for a box X^* with $\omega(X^*) < \sqrt{\varepsilon/\alpha}$ are exactly those ε-efficient points from Lemma 4.10. □

THEOREM 4.12. *Let $\mathcal{A}$ be the set generated by Algorithm* 4. *Then $\mathcal{A}$ is an (ε, δ)-efficient set of* (P).

Proof. With Lemma 4.11 we know that $\mathcal{A}$ contains only ε-efficient points.

Let $x^* \in X_E$ be efficient for (P). With Lemma 4.4 we know that a box X^* which contains x^* cannot be discarded in any while-loop. Thus, choose now a box $X^* \in \mathcal{L}_{S,3}$ with $x^* \in X^*$. This box exists because the algorithm terminates; see Lemmas 4.1 to 4.3. In lines 24 and 25 of Algorithm 4, X^* is stored in $\mathcal{L}_{S,3}$ and an ε-efficient point $x \in \mathcal{X} \cap X^*$ is added to $\mathcal{A}$. Moreover, $\omega(X^*) \leq \delta$ and thus $\|x - x^*\| \leq \delta$. □

5. Handling of constraints.

In the following we explain which difficulties arise if we are considering MOPs like (MOP) with convex inequalities described by $g_r : \mathbb{R}^n \to \mathbb{R},\ r = 1, \ldots, p$. The first issue is the handling of the list $\mathcal{L}_{PNS}$. As we explained after Algorithm 2, every point of $\mathcal{L}_{PNS}$ has to be an image of a feasible point of (MOP). This feasibility is already ensured by choosing the images of a minimal solution of the minimization of the convex underestimator over the feasible set $M^* := \{x \in X^* \mid g(x) \leq 0\}$.

The new discarding test can easily be applied to convex constraints by replacing X^0 by M, which is the feasible set given by the convex constraints and by X^0. Moreover, we have to replace X^* by M^* in every optimization problem which has to be solved. In particular, this is the case when the ideal point of a convex underestimator is determined and for solving $(P_{\bar{p},X^*})$. However, this may result in an empty feasible set M^*. In the case in which the feasible set is empty, we can discard the box. Numerically it might be difficult to verify that the feasible set is indeed empty. A quickly evaluable and efficient sufficient condition was proposed in [15] and uses interval arithmetic to obtain lower and upper bounds of g_r, $r = 1, \ldots, p$. For dealing with nonconvex constraints and finding feasible points for such optimization problems, we refer the reader to [25]. Furthermore, in [11] an approach is presented which uses the optimal solution of the dual problem to identify infeasible boxes.

6. Numerical results.

In this section we illustrate our algorithm on some test instances from the literature. We also solve an application problem which arises in engineering in the context of Lorentz force velocimetry. Algorithm 4 has been implemented in MATLAB R2017a and uses the toolbox Intlab [38] for interval arithmetic. All experiments have been done on a computer with Intel(R) Core(TM) i3-2015 CPU and 16 Gbytes RAM on operation system Windows 7 Professional.

6.1. Test instances.

First, we compare cases (II) and (III) from the beginning of subsection 3.1. Recall that (II) uses the ideal point of the convex underestimators to obtain lower bounds and compares them with $\mathcal{L}_{PNS}$ only. Case (III) implements the new discarding test. For a better comparability we restrict our algorithm to the first `while`-loop with a termination rule $\omega(X^l) < \delta$. Therefore, there is no dependency on ε in this test run.

Test instance 6.1. *This test instance is based on* [17] *and the dimension of the preimage space $n \in \mathbb{N}$ can be arbitrarily chosen:*

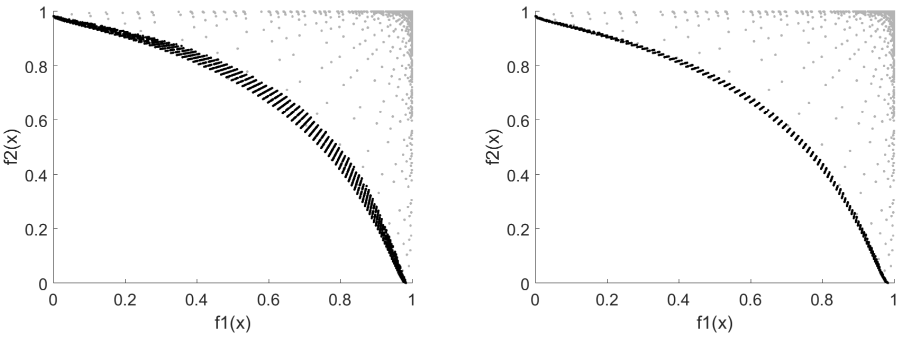
$$f(x) = \begin{pmatrix} 1 - \exp\left(-\sum\limits_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ 1 - \exp\left(-\sum\limits_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \end{pmatrix} \quad \text{with } X^0 = \left[\begin{pmatrix} -2 \\ \vdots \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ \vdots \\ 2 \end{pmatrix}\right] \in \mathbb{IR}^n.$$

For the number of iterations, the computation time t, and the number of boxes in the solution list $\mathcal{L}_S$, we obtain the results presented in Table 1.

Table 1
Results for Test instance 6.1 with $\delta = 0.1$.

	Case (II): Only ideal point			Case (III): New discarding test		
n	# iterations	t $[s]$	$\lvert\mathcal{L}_S\rvert$	# iterations	t $[s]$	$\lvert\mathcal{L}_S\rvert$
1	41	3.5970	34	41	4.1928	34
2	456	39.4317	262	359	38.1401	210
3	6283	626.7255	3434	3055	364.1089	1268
4	78965	1.0014e+04	42540	20966	2.9587e+03	7644

The illustrations in Figure 7 show the results in the image space for $n = 3$. The image set is represented by some image points in grey, which are obtained by a discretization of the feasible set X^0. The black points are the images of the midpoints of the boxes of the list $\mathcal{L}_S$.



(a) *Image set with discarding test based on case* (II).

(b) *Image set with discarding test based on case* (III).

FIG. 7. *Test instance* 6.1 *with* $n = 3,\ \delta = 0.1$.

It can be seen that we can decrease the number of iterations with the new procedure, which is clear, because computing the ideal point by minimizing convex underestimators is also a part of our discarding test (III). Additionally, for $n \geq 2$ the new procedure is faster even though we have to solve more optimization problems on each subbox. In the $n = 3$ case the approximation of the nondominated set obtained with the new discarding test is much tighter than the one which only uses the ideal point. This can be seen in Figure 7. In the other cases for n we obtained similar results.

To illustrate the whole procedure with the new discarding test and with all three `while`-loops, we chose $\varepsilon = 0.05$ and $\delta = 0.1$. The plots in Figures 8(a) and 8(b) show the partitioning of the feasible set after the second and third `while`-loops. Medium grey boxes are those which have been discarded in the first `while`-loop. The light grey boxes were discarded after the second `while`-loop. The dark grey boxes were not discarded after the second and third `while`-loops. Furthermore, the new light grey boxes compared to Figure 8(a) were discarded within the third `while`-loop. In Figure 8(c) the boxes of $\mathcal{L}_{S,3}$ are shown together with some black points which are the points from the (ε, δ)-efficient set $\mathcal{A}$. Figure 8(d) shows the image set of the test function. The black points are the images of the approximation set $\mathcal{A}$. Additionally, in Figures 8(e) and 8(f) the obtained set T is shown.

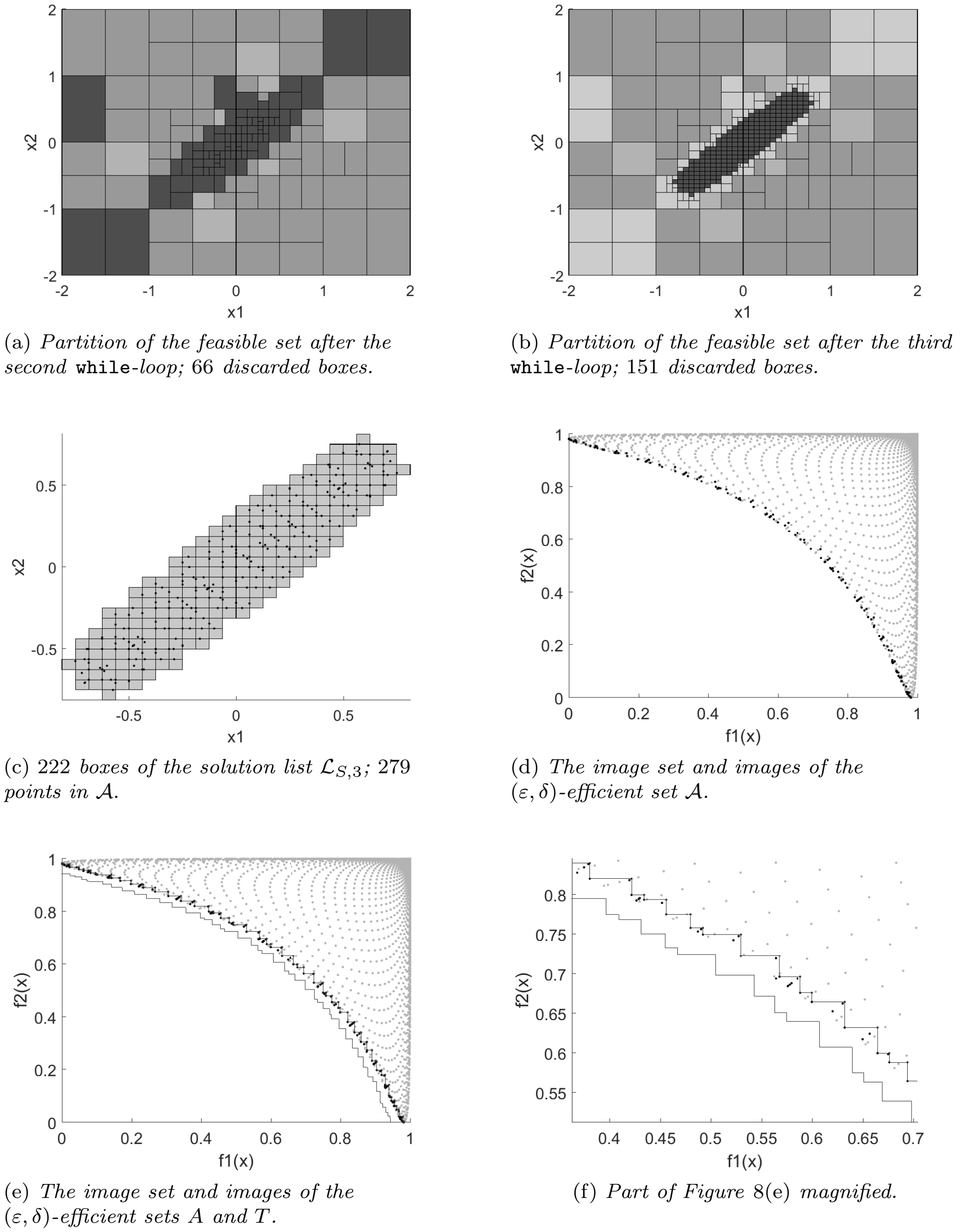
The results of the next test instance show that our new algorithm is also able to find globally efficient and nondominated points in the case in which there are also locally efficient points which are not globally efficient.

TEST INSTANCE 6.2. *This test instance was proposed in* [9]*:*

$$f(x) = \left(\begin{array}{c} x_1 \\ \frac{1}{x_1}\left(2 - \exp(-(\frac{x_2-0.2}{0.004})^2) - 0.8\exp(-(\frac{x_2-0.6}{0.4})^2)\right) \end{array} \right),$$

with $X^0 = [\binom{0.1}{0}, \binom{1}{1}] \in \mathbb{R}^2$. *The globally efficient points are* $(\tilde{x}_1, \tilde{x}_2)$ *with* $\tilde{x}_2 \approx 0.2$ *and* $\tilde{x}_1 \in [0.1, 1]$. *This test instance has also locally efficient points with* $\tilde{x}_2 \approx 0.6$ *and* $\tilde{x}_1 \in [0.1, 1]$.

Figure 9 shows the results of our algorithm on this test instance. In total 648 ε-efficient points are found, where 608 are in a δ-neighborhood of a globally efficient point. The others are ε-efficient points with $x_1 \approx 0.1$. Recall that the images of these



(a) *Partition of the feasible set after the second* `while`*-loop;* 66 *discarded boxes.*

(b) *Partition of the feasible set after the third* `while`*-loop;* 151 *discarded boxes.*

(c) 222 *boxes of the solution list* $\mathcal{L}_{S,3}$*;* 279 *points in* $\mathcal{A}$.

(d) *The image set and images of the* (ε, δ)*-efficient set* $\mathcal{A}$.

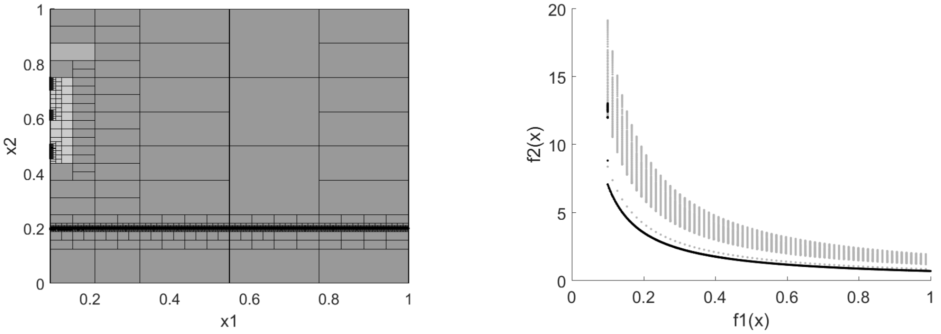
(e) *The image set and images of the* (ε, δ)*-efficient sets* A *and* T.

(f) *Part of Figure* 8(e) *magnified.*

FIG. 8. *Test instance* 6.1 *with* $n = 2$, $\varepsilon = 0.05$, *and* $\delta = 0.1$.

ε-efficient points are visualized as black points in Figure 9(b). For example a weighted sum approach with a local optimization solver such as SQP may be able to find the locally efficient points only.

The next test instance has three objective functions. Moreover, we added a convex constraint g.



(a) *Partition of the feasible set after the third* `while`*-loop;* 2039 *discarded boxes,* 648 *points in* $\mathcal{A}$.

(b) *The image set and images of the* (ε, δ)*-efficient set* $\mathcal{A}$.

FIG. 9. *Test instance* 6.2 *with* $\varepsilon = 0.01$ *and* $\delta = 0.01$.

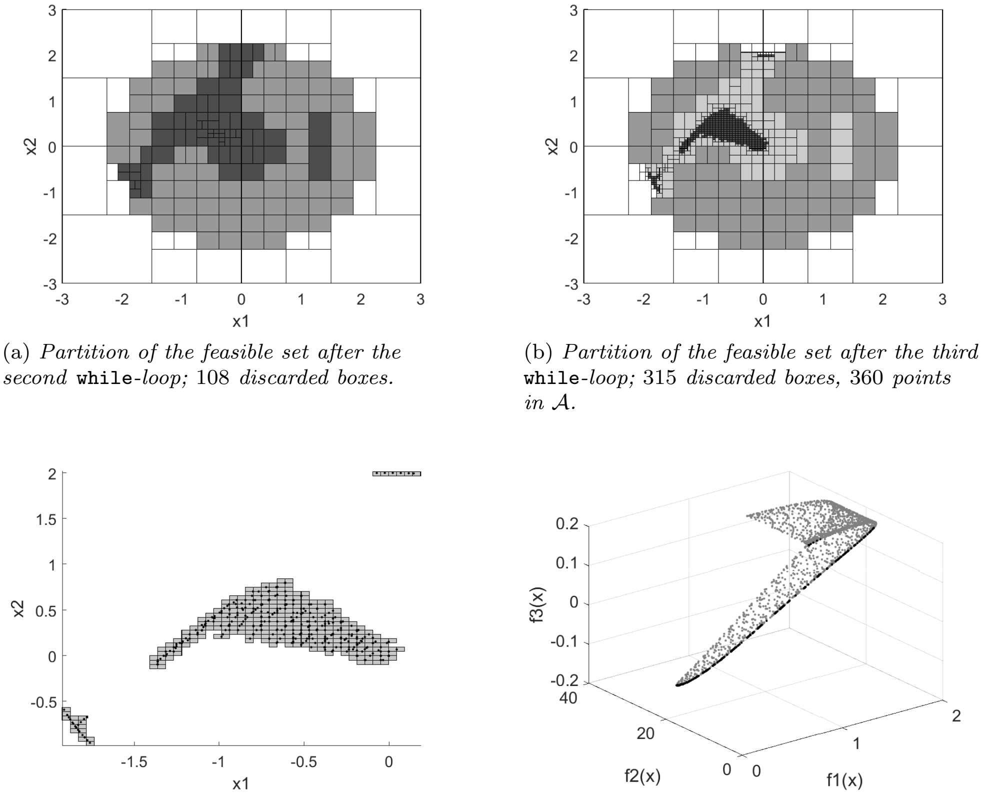
TEST INSTANCE 6.3. *This test instance (without the convex constraint) was introduced in* [43]*:*

$$f(x) = \begin{pmatrix} 0.5(x_1^2 + x_2^2)^2 + \sin(x_1^1 + x_2^2) \\ \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\ \frac{1}{x_1^2 - x_2^2 - 1} - 1.1 \exp(-x_1^2 - x_2^2) \end{pmatrix} \text{ with } X^0 = \left[\begin{pmatrix} -3 \\ -3 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix} \right] \in \mathbb{IR}^2$$

and in addition $g(x) = x_1^2 + x_2^2 - 4$.

Figure 10 shows the results of this test instance. In addition to the meanings of grey shades in the former graphical results, the white boxes are boxes which were discarded because of infeasibility. In this example no box was discarded in the second `while`-loop. Figure 10(c) shows the boxes of the solution list $\mathcal{L}_{S,3}$ and the computed ε-efficient points. Moreover, we can see in Figures 10(b) and 10(d) that in the case of disconnected areas of efficient or nondominated points our algorithm is also able to find all these areas.

6.2. Application in Lorentz force velocimetry. We have also applied our algorithm to a problem which arises in the optimization of a measurement technique known as Lorentz force velocimetry (LFV). In LFV the aim is to measure, for example, the velocity of fluids. The technique is based on measurements of the Lorentz force that occurs due to the flow of an electric conductive fluid under the influence of a variable magnetic field. The following setting was also considered in [4]. For generating a measurement system, in our problem setting n dipoles have to be arranged around a cylinder. An electric conductive fluid will flow through this cylinder during the measurements. We assume that the dipoles are placed at equidistant positions $r^i = (0, \cos\gamma_i, \sin\gamma_i)^T,\ i = 1, \dots, n,\ \gamma_i = (i-1)\frac{2\pi}{n}$ fixed, around the cylinder and aim to find the optimal magnetic orientation. We assume that all dipole moments have the same magnitude m. The orientation of an individual dipole i is then represented in terms of polar and azimuthal angles θ_i and φ_i, where φ_i is split into the two angles β_i and fixed γ_i. Thus, the magnetic moment vector m^i is $m^i(\theta_i, \beta_i) = (m\cos\theta_i, m\sin\theta_i\cos(\beta_i + \gamma_i), m\sin\theta_i\sin(\beta_i + \gamma_i))$.



(a) *Partition of the feasible set after the second* `while`*-loop;* 108 *discarded boxes.*

(b) *Partition of the feasible set after the third* `while`*-loop;* 315 *discarded boxes,* 360 *points in* $\mathcal{A}$*.*

(c) 299 *boxes of the solution list* $\mathcal{L}_{S,3}$.

(d) *The image set and images of the* (ε, δ)*-efficient set* $\mathcal{A}$*.*

FIG. 10. *Test instance* 6.3 *with* $\varepsilon = 0.1$ *and* $\delta = 0.1$.

The first objective is to maximize the absolute value of the axial force component as in [4]. Since the dipoles are on a circle of radius H in the plane $x = 0$ the force can be expressed analytically. The self-interaction term for a dipole i and the mutual interaction term between dipoles i and j are

$$F_s(\beta_i, \theta_i) = \frac{45\pi}{4096} \cdot \frac{v\sigma R^4 \mu_0^2 m^2}{128\pi H^7}[355 + 25\cos(2\theta_i) + 266\cos(2\beta_i)\sin^2\theta_i] \text{ and}$$

$$\begin{aligned} F_m(\beta_i, \beta_j, \theta_i, \theta_j) = \frac{45\pi}{1024} \cdot \frac{v\sigma R^4 \mu_0^2 m^2}{128\pi H^7} & [10(5 + 14\cos(\gamma_i - \gamma_j))\cos\theta_i\cos\theta_j \\ & + \sin\theta_i\sin\theta_j(49\cos(\gamma_i - \gamma_j - \beta_i - \beta_j) + 35\cos(\beta_i - \beta_j) \\ & + 25\cos(\gamma_i - \gamma_j + \beta_i - \beta_j) + 105\cos(\gamma_i - \gamma_j - \beta_i + \beta_j) \\ & + 35\cos(\beta_i + \beta_j) + 49\cos(\gamma_i - \gamma_j + \beta_i + \beta_j))], \end{aligned}$$

respectively. For the resulting interaction force we obtain

$$F_x(\beta, \theta) = \sum_{i=1}^{n} F_s(\beta_i, \theta_i) + \sum_{i<j} F_m(\beta_i, \beta_j, \theta_i, \theta_j).$$

The constants are the velocity of the fluid v, the electric conductivity σ, the radius of the cylinder R, and the vacuum permeability $\mu_0 = 4\pi \cdot 10^{-7}$ Vs/Am.

The second objective is a minimization of the interaction potential energy between the dipoles. The reason for that is that arrangements with a high interaction potential energy are more difficult to realize. The vector $r_{i,j}$ represents the vector between the positions of both dipoles: $r_{i,j} = r^j - r^i$. The function is the sum of all energies between every pair of dipoles:

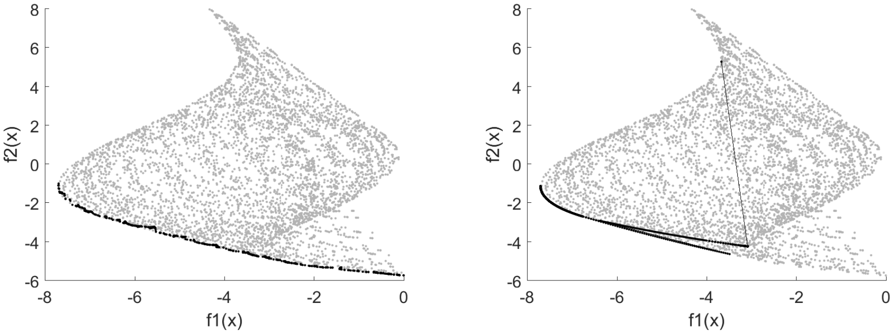
$$V(\beta,\theta) = \sum_{i<j} \frac{\mu_0}{4\pi |r_{i,j}|^5} [|r_{i,j}|^2 m_i(\theta_i,\beta_i) m_j(\theta_j,\beta_j) - 3(m_i(\theta_i,\beta_i) r_{i,j})(m_j(\theta_j,\beta_j) r_{i,j})].$$

To reduce the dimension of the decision space we fix $\theta_i = \frac{\pi}{2}$ for all $i = 1,\dots,n$. Because of this and the symmetry of the arrangements, the feasible set of interesting angles β is given by

$$X^0 = \left[\left(-\frac{\pi}{2}, -\pi, \dots, -\pi\right)^T, \left(\frac{\pi}{2}, \pi, \dots, \pi\right)^T \right] \in \mathbb{R}^n.$$

All constant coefficients are set to 1 and we scale the objective functions by -0.1 (to switch from maximization to minimization) and 100, respectively, to obtain different accuracies for both objective functions.

The following plot in Figure 11(a) shows the image set after executing the algorithm with $\varepsilon = \delta = 0.5$ for three dipoles. Figure 11(b) shows some results after using a weighted sum approach with a standard SQP solver for the scalarized problems. The black points connected by thin black lines are the minimal solutions of minimizing a weighted sum of the objectives and are also used as the starting point for the next weighted sum. Obviously, this approach was able to find locally efficient points first and only after some iterations some of the globally efficient points. Moreover, the lower-right part of the nondominated set was not found. We tried different starting points, but never obtained an approximation of the whole nondominated set, as was achieved with our algorithm in Figure 11(a).



(a) *The image set after Algorithm* 4.

(b) *The image set after a weighted sum approach with a random starting point.*

FIG. 11. *Graphical results for* 3 *dipoles on a circle around a cylinder.*

7. Conclusions.

We have combined the ideas of convex underestimators with techniques from multiobjective convex optimization and the idea of local upper bounds from multiobjective combinatorial optimization to obtain an efficient discarding test. In contrast to other global multiobjective optimization algorithms, our algorithm guarantees the (ε, δ)-efficiency of the output $\mathcal{A}$ in a finite time. Because of this, the introduced algorithm is not comparable with existing algorithms. Numerical experiments show that good results can be obtained even with large values for ε and δ. This is due to the fact that the estimations in some proofs are quite rough.

The current implementation does not use other, simpler, discarding tests as already proposed in the literature. Therefore, further time savings can be expected if these existing discarding tests are used additionally. An example of these tests are the monotonicity tests as proposed in [15, 40]. Moreover, in [45] interesting bounds based on Lipschitz constants are derived and it would be of interest to explore possible combinations. Also, the handling of nonconvex constraints should be considered. However, the same difficulties as for global single objective optimization algorithms will arise. These difficulties are discussed, for instance, in [25]. Another generalization of the proposed algorithm to mixed integer problems using also ideas from [6] is a work in progress.

Acknowledgments. The authors wish to thank Prof. Dr. A. Löhne and Dr. B. Weißing (Friedrich-Schiller-Universität Jena) as well as Prof. Dr. K. Klamroth (Bergische Universität Wuppertal) for valuable discussions. Moreover, we thank PD Dr. T. Boeck (TU Ilmenau) for the cooperation concerning Lorentz force velocimetry and M.Sc. S. Rocktäschel (TU Ilmenau) for supporting the implementations. We would like to thank the two anonymous referees for their careful reading and constructive remarks, which helped to improve this paper significantly.

REFERENCES

[1] C. Adjiman, S. Dallwig, C. Floudas, and A. Neumaier, *A global optimization method, αBB, for general twice-differentiable constrained NLPs:* I – *Theoretical advances*, Comput. Chem. Eng., 22 (1998), pp. 1137–1158.

[2] C. Audet, P. Hansen, B. Jaumard, and G. Savard, *A branch and cut algorithm for nonconvex quadratically constrained quadratic programming*, Math. Program., 87 (2000), pp. 131–152.

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

[4] T. Boeck, D. Terzijska, and G. Eichfelder, *Maximum electromagnetic drag configurations for a translating conducting cylinder with distant magnetic dipoles*, J. Engrg. Math., 108 (2018), pp. 123–141.

[5] H. Bonnel, A. Iusem, and B. Svaiter, *Proximal methods in vector optimization*, SIAM J. Optim., 15 (2005), pp. 953–970.

[6] C. Buchheim, M. De Santis, S. Lucidi, F. Rinaldi, and L. Trieu, *A feasible active set method with reoptimization for convex quadratic mixed-integer programming*, SIAM J. Optim., 26 (2016), pp. 1695–1714.

[7] E. Campana, M. Diez, G. Liuzzi, S. Lucidi, R. Pellegrini, V. Piccialli, F. Rinaldi, and A. Serani, *A multi-objective DIRECT algorithm for ship hull optimization*, Comput. Optim. Appl., 71 (2018), pp. 53–72.

[8] A. Custódio and J. Madeira, *MultiGLODS: Global and local multiobjective optimization using direct search*, J. Global Optim., 72 (2018), pp. 323–345.

[9] K. Deb, *Multi-objective genetic algorithms: Problem difficulties and construction of test problems*, Evol. Comput., 7 (1999), pp. 205–230.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA*-II, IEEE Trans. Evol. Comput., 6 (2002), pp. 181–197.

[11] M. Dür, *Dual bounding procedures lead to convergent branch-and-bound algorithms*, Math. Program., 91 (2001), pp. 117–125.

[12] M. Ehrgott, *Multicriteria Optimization*, Springer, New York, 2005.

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

[14] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer, New

York, 2008.

[15] J. Fernández and B. Tóth, *Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods*, Comput. Optim. Appl., 42 (2009), pp. 393–419.

[16] J. Fliege and A. F. Vaz, *A method for constrained multiobjective optimization based on SQP techniques*, SIAM J. Optim., 26 (2016), pp. 2091–2119.

[17] C. Fonseca and P. Fleming, *Multiobjective genetic algorithms made easy: Selection sharing and mating restriction*, in Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, IEEE Press, Piscataway, NJ, 1995, pp. 45–52.

[18] D. Gourion and D. Luc, *Finding efficient solutions by free disposal outer approximation*, SIAM J. Optim., 20 (2010), pp. 2939–2958.

[19] C. Gutiérrez, B. Jiménez, and V. Novo, *A unified approach and optimality conditions for approximate solutions of vector optimization problems*, SIAM J. Optim., 17 (2006), pp. 688–710.

[20] W. W. Hager and D. T. Phan, *An ellipsoidal branch and bound algorithm for global optimization*, SIAM J. Optim., 20 (2009), pp. 740–758.

[21] E. Hendrix and B. Tóth, *Introduction to Nonlinear and Global Optimization*, Springer, New York, 2010.

[22] J. Jahn, *Multiobjective search algorithm with subdivision technique*, Comput. Optim. Appl., 35 (2006), pp. 161–175.

[23] J. Jahn, *Vector Optimization: Theory, Applications, and Extensions*, Springer, New York, 2011.

[24] D. Jones, C. Perttunen, and B. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, J. Optim. Theory Appl., 79 (1993), pp. 157–181.

[25] P. Kirst, O. Stein, and P. Steuermann, *Deterministic upper bounds for spatial branch-and-bound methods in global minimization with nonconvex constraints*, TOP, 23 (2015), pp. 591–616.

[26] K. Klamroth, *Private communication*, 2017.

[27] K. Klamroth, R. Lacour, and D. Vanderpooten, *On the representation of the search region in multi-objective optimization*, European J. Oper. Res., 245 (2015), pp. 767–778.

[28] S. Kutateladze, *Convex ε-programming*, Soviet Math. Dokl., 20 (1979), pp. 391–393.

[29] D. Lera and Y. Sergeyev, *Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives*, SIAM J. Optim., 23 (2013), pp. 508–529.

[30] M. Locatelli and F. Schoen, *Global Optimization: Theory, Algorithms, and Applications*, MOS-SIAM Ser. Optim. 15, SIAM, Philadelphia, PA, 2013.

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

[32] P. Loridan, *ε-solutions in vector minimization problems*, J. Optim. Theory Appl., 43 (1984), pp. 265–276.

[33] A. Lovison, *Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis*, SIAM J. Optim., 21 (2011), pp. 463–490.

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

[35] D. Mueller-Gritschneder, H. Graeb, and U. Schlichtmann, *A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems*, SIAM J. Optim., 20 (2009), pp. 915–934.

[36] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.

[37] P. Pardalos, A. Žilinskas, and J. Žilinskas, *Non-convex Multi-objective Optimization*, Springer, New York, 2017.

[38] S. Rump, *INTLAB – INTerval LABoratory*, in Developments in Reliable Computing, T. Csendes, ed., Springer, New York, 1999, pp. 77–104.

[39] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*, Academic Press, New York, 1985.

[40] D. Scholz, *Deterministic Global Optimization: Geometric Branch-and-Bound Methods and Their Applications*, Springer, New York, 2012.

[41] M. Schulze Darup and M. Mönnigmann, *Improved automatic computation of Hessian matrix spectral bounds*, SIAM J. Sci. Comput., 38 (2016), pp. A2068–A2090.

[42] H. Tuy and R. Horst, *Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and d.c. optimization problems*, Math.

Program., 41 (1988), pp. 161–183.

[43] R. Viennet, C. Fonteix, and I. Marc, *Multicriteria optimization using a genetic algorithm for determining a Pareto set*, Internat. J. Systems Sci., 27 (1996), pp. 255–260.

[44] L. Wang, H. Ishida, T. Hiroyasu, and M. Miki, *Examination of multi-objective optimization method for global search using DIRECT and GA*, in Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE Press, Piscataway, NJ, 2008, pp. 2446–2451.

[45] A. Žilinskas and J. Žilinskas, *Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems*, Commun. Nonlinear Sci. Numer. Simul., 21 (2016), pp. 89–98.