Advancements in the computation of enclosures for multi-objective optimization problems

Gabriele Eichfelder*, Leo Warnow*

February 25, 2023

Abstract

A central goal for multi-objective optimization problems is to compute their nondominated sets. In most cases these sets consist of infinitely many points and it is not a practical approach to compute them exactly. One solution to overcome this problem is to compute an enclosure, a special kind of coverage, of the nondominated set. For that computation one often makes use of so-called local upper bounds. In this paper we present a generalization of this concept. For the first time, this allows to apply a warm start strategy to the computation of an enclosure. We also show how this generalized concept allows to remove empty areas of an enclosure by deleting certain parts of the lower and upper bound sets which has not been possible in the past. We demonstrate how to apply our ideas to the box approximation algorithm, a general framework to compute an enclosure, as recently used in the solver called BAMOP. We show how that framework can be simplified and improved significantly, especially concerning its practical numerical use. In fact, we show for selected numerical instances that our new approach is up to eight times faster than the original one. Hence, our new framework is not only of theoretical but also of practical use, for instance for continuous convex or mixed-integer quadratic optimization problems.

Key Words: multi-objective optimization, nonlinear optimization, mixed-integer optimization, enclosure, warm start strategies

Mathematics subject classifications (MSC 2010): 90C11, 90C26, 90C29, 90C30, 90C59

1 Introduction

In multi-objective optimization multiple objective functions are minimized simultaneously. Those objective functions are, in general, conflicting. As a result, typically there exists no feasible point that minimizes all objective functions at the same time. Hence, the optimality concepts of efficient points (in the decision space) and nondominated points (in the criterion space) are used. Since usually the set of nondominated points is not finite, a typical goal in multi-objective optimization is to compute an approximation of this set. This is also the aim of this paper.

*Institute of Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, {gabriele.eichfelder,leo.warnow}@tu-ilmenau.de

For continuous objective functions $f_i\colon \mathbb{R}^n \to \mathbb{R}$, $i \in [m] := \{1, 2, \dots, m\}$, and a nonempty and compact feasible set $S \subseteq \mathbb{R}^n$ we consider the multi-objective optimization problem

$$\min_x \ f(x) \quad \text{s.t.} \quad x \in S \tag{MOP}$$

where $f = (f_1, f_2, \dots, f_m)\colon \mathbb{R}^n \to \mathbb{R}^m$. All theoretical results in this paper need no further assumptions than continuous objective functions and a compact feasible set. However, the practical applicability of the proposed solution algorithm depends on the availability of fast and reliable solvers for single-objective subproblems derived from (MOP), see (SUP(l, u)) on page 14. By now this includes problem classes such as multi-objective continuous convex optimization problems or multi-objective mixed-integer quadratic optimization problems.

As already mentioned, our aim is to compute an approximation of the nondominated set of (MOP) denoted by $\mathcal{N}$. More precisely, we will compute a special kind of coverage of the nondominated set, called enclosure. This is basically a union of boxes given by a lower bound set $L \subseteq \mathbb{R}^m$ and an upper bound set $U \subseteq \mathbb{R}^m$ with $\mathcal{N} \subseteq (L + \mathbb{R}^m_+) \cap (U - \mathbb{R}^m_+)$. One method to compute an enclosure for general multi-objective continuous nonconvex optimization problems has been presented in [11]. Very recently, this approach has been generalized to the mixed-integer setting in [12]. The algorithms suggested in [11] and [12] are spatial branch-and-bound methods. While the branch-and-bound approach enables them to solve general nonconvex optimization problems, it also limits their performance for increasing dimension of the decision space. The enclosure concept, and hence the termination criterion of these algorithms, works entirely in the criterion space. This motivates the development of solution approaches that may only work for a smaller class of optimization problems, but are more independent of decision space constructions.

For multi-objective mixed-integer convex optimization problems such an approach has been presented with the HyPaD algorithm in [14]. This algorithm works almost entirely in the criterion space and does not use any decision-space-based techniques such as branch-and-bound. It exploits that a mixed-integer optimization problem can be decomposed into a family of purely continuous optimization problems by fixing the assignments of the integer variables. This also implies that a coverage of the overall nondominated set can be obtained by combining coverages of the nondominated sets of those continuous subproblems. While several strategies to reduce the number of subproblems that need to be considered to obtain an enclosure of the nondominated set of the original multi-objective mixed-integer problem are presented in [14], in the worst case the HyPaD algorithm needs to explore all possible integer assignments and the corresponding subproblems.

In this paper, we present a new algorithm that extends and combines some of the ideas from [14] and from another enclosure algorithm from [13] called BAMOP. In [13] the same optimization problem (MOP) as in this paper has been discussed. Hence, BAMOP can be applied to exactly the same optimization problems as the algorithm we are going to present in this paper. In [13] the authors mainly focus on multi-objective continuous convex optimization problems, in particular for their numerical tests. However, it can also be used for further problem classes such as multi-objective mixed-integer quadratic optimization problems. Moreover, by incorporating some of the ideas from [14], some of its subroutines can be simplified and its performance can be improved significantly. Thus, our new algorithm can also be interpreted as a vastly improved and generalized version of BAMOP. For the specific case of multi-

objective mixed-integer problems it also overcomes the drawback of HyPaD which needs to explore continuous subproblems obtained by fixing the assignment of integer variables.

All of the above mentioned algorithms [11, 12, 13, 14] to compute an enclosure make use of the bound concepts from [21]. In that paper the initial coverage of the nondominated set is a single box $B \subseteq \mathbb{R}^m$ which is then iteratively improved. It was already discussed in [13] that one of the big advantages of using an enclosure (or box coverage) is that boxes respect the natural ordering. While this is also true for some other approximation approaches, for instance the one presented in [24] for multi-objective nonconvex optimization problems, it is not the case for general sandwiching techniques (e.g., [6, 8, 22, 29]). For instance, in case of multi-objective mixed-integer optimization problems, enclosures allow to combine approximations obtained for continuous subproblems that arise when fixing the integer assignments. However, if one wants to further improve such enclosures that are the result of combining approximations for certain subproblems, one no longer starts with a single box B. Instead, one starts with a union of boxes. In this paper we show how to extend the bound concepts from [21] in order to allow the computation of enclosures even if the initial enclosure is not just a single box. In particular, the presented theory for more general initializations of an enclosure, which we also refer to as warm start strategies, can not only be used within our algorithm but within all algorithms that make use of the concepts from [21], including [11, 12, 13, 14].

The concept of local upper bounds from [21], which is related to a bound concept that appeared earlier in [7], is widely used in the literature. Besides for the above mentioned enclosure algorithms, it also appears within the context of hypervolume based approaches as in [25, 31]. It has also been used in order to compute a representation of the nondominated set (instead of a coverage) in [5]. Next to that, there is ongoing research on how to compute the so-called search regions, which are closely related to the local upper bounds (see Section 3), for example in [4] and more recently in [19].

The remaining paper is structured as follows. In Section 2 we briefly present the most important notations and definitions that are used within this paper. In Section 3 we give a formal definition of the above mentioned local upper bound concept and recap its use within enclosure algorithms from the literature. We also present the first warm start strategies in that section. We extend the local upper bound concept in Section 4 and show how this generalized bound concept allows for advanced warm start strategies and improvements in the computation of an enclosure. Then, in Section 5, we combine the theoretical results and present our Advanced Enclosure Algorithm (AdEnA). Finally, in Section 6, we present the numerical results of our algorithm for selected test instances and compare it with other algorithms from the literature such as HyPaD for multi-objective mixed-integer convex quadratic optimization problems and BAMOP for multi-objective continuous convex optimization problems.

2 Notations and Definitions

All relations, e.g., $x^1 \leq x^2$ for $x^1, x^2 \in \mathbb{R}^n$ are meant to be read componentwise. We denote the (closed) box with lower bound $l \in \mathbb{R}^m$ and upper bound $u \in \mathbb{R}^m$ by $[l,u] := \{y \in \mathbb{R}^m \mid l \leq y \leq u\}$ and the corresponding open box by $(l,u) := \{y \in \mathbb{R}^m \mid l < y < u\}$.
Since usually the objective functions of (MOP) are competing with each other, in general, it is not possible to find a feasible point that minimizes all objectives at the same time. Thus, we use the optimality concepts of efficiency and nondominance.

Definition 2.1 *A point $\bar{x} \in S$ is called an efficient solution of (MOP) if there exists no $x \in S$ with $f_i(x) \leq f_i(\bar{x})$ for all $i \in [m]$ and $f_j(x) < f_j(\bar{x})$ for at least one $j \in [m]$. It is called a weakly efficient solution of (MOP) if there exists no $x \in S$ with $f_i(x) < f_i(\bar{x})$ for all $i \in [m]$.*

Since our algorithm works almost entirely in the criterion space, we need a corresponding concept there as well, which is the concept of nondominated points.

Definition 2.2 *Let $y^1, y^2 \in \mathbb{R}^m$. The point y^2 is dominated by y^1 if $y^1 \neq y^2$, $y^1 \leq y^2$. For a set $N \subseteq \mathbb{R}^m$ a point $y \in \mathbb{R}^m$ is dominated given N if there exists $\hat{y} \in N$ such that $\hat{y} \neq y$, $\hat{y} \leq y$. If y is not dominated given N, it is called nondominated given N and in case $y \in N$ it is called a nondominated point of N.*
Analogously, the point y^2 is strictly dominated by y^1 if $y^1 < y^2$ and a point $y \in \mathbb{R}^m$ is strictly dominated given a set $N \subseteq \mathbb{R}^m$ if there exists $\hat{y} \in N$ such that $\hat{y} < y$. If y is not strictly dominated given N, it is called weakly nondominated given N and in case $y \in N$ it is called a weakly nondominated point of N.

Since the images $f(\bar{x})$ of efficient solutions $\bar{x} \in S$ of (MOP) are nondominated given $f(S)$, they are called nondominated points of (MOP). We denote by $\mathcal{E} \subseteq \mathbb{R}^n$ the set of efficient solutions (also efficient set) and by $\mathcal{N} \subseteq \mathbb{R}^m$ the set of nondominated points (also nondominated set) of (MOP), i.e., $\mathcal{N} := \{f(x) \in \mathbb{R}^m \mid x \in \mathcal{E}\}$.
The aim of this paper is to compute a specific approximation, called enclosure, of the nondominated set $\mathcal{N}$ of (MOP). While it was defined for general nonempty and compact sets in [11], we use it here as in [13, 14] for finite sets only and adapt the definition accordingly.

Definition 2.3 *Let $L, U \subseteq \mathbb{R}^m$ be two finite sets with $\mathcal{N} \subseteq L + \mathbb{R}^m_+$ and $\mathcal{N} \subseteq U - \mathbb{R}^m_+$. Then L is called lower bound set, U is called upper bound set, and the set $\mathcal{A}$ which is given as*

$$\mathcal{A} = \mathcal{A}(L,U) := (L + \mathbb{R}^m_+) \cap (U - \mathbb{R}^m_+) = \bigcup_{l \in L} \bigcup_{\substack{u \in U, \\ l \leq u}} [l,u]$$

is called enclosure (or box approximation) of the nondominated set $\mathcal{N}$ of (MOP) given L and U.

The quality of an enclosure $\mathcal{A}$, which is used as termination criterion for its computation, is typically given by its width $w(\mathcal{A})$. It is also presented in [11] and defined as the optimal value of

$$\max_{l,u} \; s(l,u) \quad \text{s.t.} \quad l \in L,\; u \in U,\; l \leq u \tag{2.1}$$

where $s(l,u) := \min\{u_i - l_i \mid i \in [m]\}$ denotes the shortest edge length of a box $[l,u]$. A more detailed discussion and extensive motivation for this quality measure is provided in [11, 13].

3 Classic local upper and local lower bounds

The computation of an enclosure mainly depends on the computation of the bound sets $L, U \subseteq \mathbb{R}^m$. A commonly used approach in the literature, e.g., in [13] and [14], is to make use of the so-called local upper bounds (LUBs) for this. This concept has been presented and extensively discussed in [21], where it was only used for the computation of an upper bound set. However, it can be extended for the computation of a lower bound set as well.
In this section we present what we refer to as the classic local upper bound concept. This is basically the concept as it was presented in [21], but with a slightly changed notation to be able to use it not only for upper but also for lower bounds. This notation has also been used in [14].
A generalization of local upper bounds (and local lower bounds) will be presented in the next section. This generalization allows for various improvements concerning the computation of an enclosure, including warm start strategies. However, since the classic local upper bound and local lower bound concepts are widely used within the literature, e.g., in [11, 12, 13, 14], and some warm start techniques are already possible without changing these concepts, we introduce those first.
As a prerequisite we need a definition of so-called stable sets. According to [21] a set $Y \subseteq \mathbb{R}^m$ is called stable if the elements of Y are not pairwise comparable, this is, for all $y^1, y^2 \in Y, y^1 \neq y^2$ there exist $i, j \in [m]$ such that $y^1_i < y^2_i$ and $y^1_j > y^2_j$. Further, for $z, Z \in \mathbb{R}^m$ we denote by $B := [z, Z] = \{y \in \mathbb{R}^m \mid z \leq y \leq Z\}$ a closed box, such that $f(S) \subseteq \operatorname{int}(B) = (z, Z)$. Due to our assumptions it is guaranteed that such a box B always exists.

Definition 3.1 *Let $N \subseteq \operatorname{int}(B)$ be a finite and stable set. Then the lower search region for N is $s(N) := \operatorname{int}(B) \setminus (N + \mathbb{R}^m_+)$ and the lower search zone for some $u \in \mathbb{R}^m$ is $c(u) := \{y \in \operatorname{int}(B) \mid y < u\}$. A set $U = U(N) \subseteq B$ is called local upper bound set given N if*

(i) $s(N) = \bigcup_{u \in U(N)} c(u)$,

(ii) $c(u^1) \not\subseteq c(u^2)$ *for all* $u^1, u^2 \in U(N)$, $u^1 \neq u^2$.

Each point $u \in U(N)$ is called a local upper bound (LUB).

We remark that in case $N \subseteq f(S)$ the lower search region $s(N)$ can indeed be interpreted as the area where to "search" for nondominated points of (MOP). As already mentioned, the same concept can be used to obtain so-called local lower bounds as follows.

Definition 3.2 *Let $N \subseteq \operatorname{int}(B)$ be a finite and stable set. Then the upper search region for N is $S(N) := \operatorname{int}(B) \setminus (N - \mathbb{R}^m_+)$ and the upper search zone for some $l \in \mathbb{R}^m$ is $C(l) := \{y \in \operatorname{int}(B) \mid y > l\}$. A set $L = L(N) \subseteq B$ is called local lower bound set given N if*

(i) $S(N) = \bigcup_{l \in L(N)} C(l)$,

(ii) $C(l^1) \not\subseteq C(l^2)$ *for all* $l^1, l^2 \in L(N)$, $l^1 \neq l^2$.

Each point $l \in L(N)$ is called a local lower bound (LLB).

We want to point out that for Definitions 3.1 and 3.2 it is not necessary to assume the sets N to be stable, see [21, Remark 2.2] and [14, Remark 3.5].
The following lemma is taken from [14] and shows that for certain choices of the sets $N \subseteq \operatorname{int}(B)$ local upper bounds and local lower bounds are indeed bounds in the sense of Definition 2.3.

Lemma 3.3 *Let $N^1 \subseteq f(S)$ and $N^2 \subseteq \operatorname{int}(B) \setminus (f(S) + \operatorname{int}(\mathbb{R}^m_+))$ be finite and stable. Then $U(N^1)$ is an upper bound set and $L(N^2)$ is a lower bound set in the sense of Definition 2.3.*

The general idea used to compute an enclosure of the nondominated set $\mathcal{N}$ of (MOP) in [13, 14], which serve as a basis for our new algorithm, is basically as follows. We start with $N^1 = \emptyset$, $N^2 = \emptyset$ and obtain the local upper and local lower bound sets $U(N^1) = U(\emptyset) = \{Z\}$ and $L(N^2) = L(\emptyset) = \{z\}$. Hence, the initial enclosure is $\mathcal{A}(L(N^2), U(N^1)) = [z, Z] = B$. Then, iteratively the sets N^1 and N^2 are updated by adding new points $y \in \operatorname{int}(B)$ to them, which we refer to as update points. For the update of the local upper bound set we use [21, Algorithm 3], see Algorithm 1. The same technique can be used to update the set of local lower bounds, see Algorithm 2.

Algorithm 1 Updating a local upper bound set

Input: Local upper bound set $U(N)$ and update point $y \in \mathbb{R}^m$
Output: Updated local upper bound set $U(N \cup \{y\})$

procedure UpdateLUB($U(N), y$)
  $A = \{u \in U(N) \mid y < u\}$
  for $i \in [m]$ **do**
    $B_i = \{u \in U(N) \mid y_i = u_i \text{ and } y_{-i} < u_{-i}\}$
    $P_i = \emptyset$
  end for
  for $i \in [m]$ **do**
    for $u \in A$ **do**
      $P_i = P_i \cup \{(y_i, u_{-i})\}$
    end for
  end for
  for $i \in [m]$ **do**
    $P_i = \{u \in P_i \mid u \not\leq u' \text{ for all } u' \in P_i \cup B_i,\ u' \neq u\}$
  end for
  $U(N \cup \{y\}) = (U(N) \setminus A) \cup \bigcup_{i \in [m]} P_i$
end procedure

Within both algorithms the following notation for projections from [21] is used. For $y \in \mathbb{R}^m, \alpha \in \mathbb{R}$ and an index $i \in [m]$ we define

$$\begin{aligned} y_{-i} &:= (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)^\top \text{ as well as} \\ (\alpha, y_{-i}) &:= (y_1, \ldots, y_{i-1}, \alpha, y_{i+1}, \ldots, y_m)^\top. \end{aligned}$$

The computation of the enclosure terminates when the sets N^1 and N^2 have been updated in such a way that $w(\mathcal{A}(L(N^2), U(N^1)) \leq \varepsilon$, where $\varepsilon > 0$ is a prescribed quality parameter.

Algorithm 2 Updating a local lower bound set

Input: Local lower bound set $L(N)$ and update point $y \in \mathbb{R}^m$
Output: Updated local lower bound set $L(N \cup \{y\})$

procedure UPDATELLB($L(N), y$)
  $A = \{l \in L(N) \mid y > l\}$
  for $i \in [m]$ **do**
    $B_i = \{l \in L(N) \mid y_i = l_i \text{ and } y_{-i} > l_{-i}\}$
    $P_i = \emptyset$
  end for
  for $i \in [m]$ **do**
    for $l \in A$ **do**
      $P_i = P_i \cup \{(y_i, l_{-i})\}$
    end for
  end for
  for $i \in [m]$ **do**
    $P_i = \{l \in P_i \mid l \not\geq l' \text{ for all } l' \in P_i \cup B_i,\ l' \neq l\}$
  end for
  $L(N \cup \{y\}) = (L(N) \setminus A) \cup \bigcup_{i \in [m]} P_i$
end procedure

For the remaining part of this section, we discuss advanced initialization strategies for the computation of an enclosure using the classic concept of local upper and local lower bounds as recalled above. So far, we assumed the initial enclosure to be $\mathcal{A}(L, U) = [z, Z] =: B$. In terms of Lemma 3.3 this corresponds to $N^1 = N^2 = \emptyset$ and the local upper and local lower bound sets $U = U(\emptyset) = \{Z\}$ and $L = L(\emptyset) = \{z\}$. However, any sets $N^1, N^2 \subseteq \text{int}(B)$ that satisfy the assumptions of Lemma 3.3 result in a valid initial enclosure. In the following, we will give some examples on how to obtain such sets N^1, N^2 using information about the optimization problem (MOP) that might already be available before starting the computation of an enclosure. This is why we refer to those advanced initialization strategies as warm starts.

Warm starting the upper bound set U is possible by providing any set $N^1 \subseteq f(S)$ of attainable image points. Obtaining such a set is often possible by using computationally cheap or heuristic methods. A simple strategy to warm start both lower and upper bounds simultaneously is to compute a selection $N = N^1 = N^2$ of weakly nondominated points. This can, for example, be achieved by solving scalarizations of the original problem (MOP). In particular, this allows to combine the advantages of representation approaches, i.e., such approximation algorithms that compute a finite subset of the nondominated set, with those of the enclosure concept. For example there is a clear quality measure for enclosures (their width) which is easy to evaluate. For continuous multi-objective optimization problems, corresponding scalarization techniques are presented for example in [2, 9, 16, 20, 30]. A good overview is also provided by the surveys [28] and more recently [10]. For the mixed-integer case, scalarization techniques are provided in [1].

Considering the initialization of N^2, not only weakly nondominated points of (MOP), but also weakly nondominated points of relaxations of (MOP) can be used to obtain an initial enclosure. This includes, for instance, continuous convex relaxations of mixed-integer problems that allow to use all the scalarization approaches mentioned

above. For mixed-integer linear problems also the so-called nondominated extreme points (sometimes also called extreme supported nondominated points) are valid candidates for N^2. Methods to compute such points are presented in [32] and more recently in [27]. Be aware that in order to satisfy the assumptions of Lemma 3.3, one needs to ensure $N^2 \subseteq \text{int}(B)$, which might not be the case for the nondominated points of relaxations of (MOP).
All previous examples aimed to warm start the computation of an enclosure by computing sets $N^1, N^2 \subseteq \text{int}(B)$ as in Lemma 3.3 and to initialize the enclosure as $\mathcal{A}(L, U)$ with $U = U(N^1)$ and $L = L(N^2)$ being the corresponding local upper and local lower bound sets. However, there is another warm start scenario that is also of practical relevance, but not covered by this approach. In this scenario, a lower bound set $L \subseteq \mathbb{R}^m$ and an upper bound set $U \subseteq \mathbb{R}^m$ as in Definition 2.3 are known, but it is not clear whether those are also local lower and local upper bound sets, i.e., if there exist $N^1, N^2 \subseteq \text{int}(B)$ such that $U = U(N^1)$ and $L = L(N^2)$. We illustrate this with the next example.

Example 3.4 We consider the tri-objective mixed-integer convex quadratic optimization problem

$$\begin{aligned} \min_x \ & (x_1 + x_4, x_2 + x_5, x_3 + x_6)^\top \quad \text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 \leq 1, \\ & x_4^2 + x_5^2 + x_6^2 \leq 1, \\ & x_1, x_2, x_3 \in \mathbb{R}, \\ & x_4, x_5, x_6 \in \mathbb{Z}. \end{aligned} \tag{Ex1}$$

For this problem the continuous and the integer variables are separable. Due to the constraints, the continuous variables (x_1, x_2, x_3) describe the unit ball in $\mathbb{R}^3$. This ball is attached to all points $(x_4, x_5, x_6)^\top \in \mathbb{Z}^3$, described by the integer variables, that are contained within the unit ball in $\mathbb{R}^3$. In particular, there are only three assignments of the integer variables (x_4, x_5, x_6) that contribute to the nondominated set of (Ex1). These are $(-1, 0, 0)$, $(0, -1, 0)$, and $(0, 0, -1)$. For each of these assignments we obtain a purely continuous subproblem that describes a ball with radius $r = 1$ in the criterion space, see Figure 1.

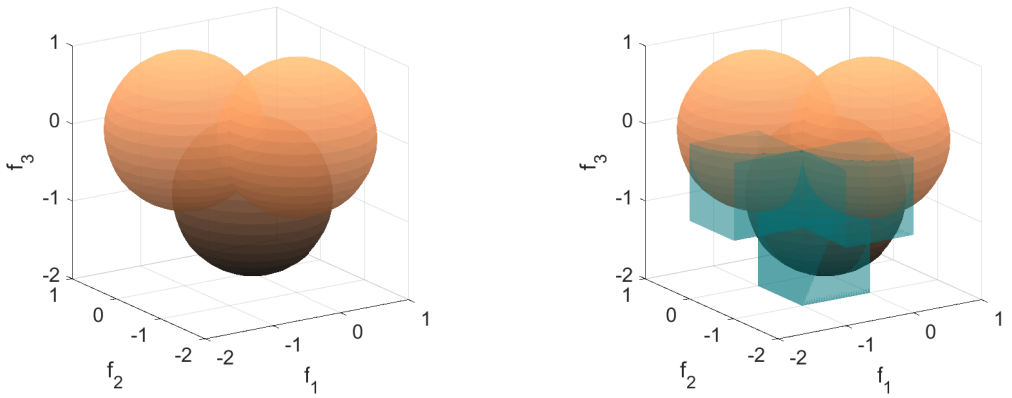

Figure 1: Image points that potentially contribute to the nondominated set of (Ex1) and a possible initial enclosure consisting of three lower and upper bounds

Hence, we know that the nondominated set $\mathcal{N}$ of (Ex1) is a subset of $L + \mathbb{R}^3_+$ for

$$L = \{(-2, -1, -1)^\top, (-1, -2, -1)^\top, (-1, -1, -2)^\top\}.$$

Further, for

$$U = \{(-1,0,0)^\top, (0,-1,0)^\top, (0,0,-1)^\top\}$$

it holds $\mathcal{N} \subseteq U - \mathbb{R}^3_+$. Thus, for this example, $\mathcal{A}(L,U)$ would be a good initial enclosure to start with.
However, L and U are not local lower and local upper bound sets in the sense of Definitions 3.1 and 3.2. To see this, assume to the contrary that there exists a finite and stable set $N \subseteq \operatorname{int}(B) = (z,Z)$ with $L = L(N)$. Thereby, $z, Z \in \mathbb{R}^m$ can be chosen arbitrarily as long as it holds $f(S) \subseteq (z,Z)$. A possible choice would be $z = (-3,-3,-3)^\top$ and $Z = (3,3,3)^\top$. By Definition 3.2 (i) we have that for all $\bar{y}_3 \in (z_3, Z_3)$ the point $\bar{y} := (-2,-2,\bar{y}_3)^\top \in \operatorname{int}(B)$ is not included in the upper search region, i.e., $\bar{y} \notin S(N)$. Thus, for all $\bar{y}_3 \in (z_3, Z_3)$ there exists some $y' \in N$ such that $\bar{y} \leq y'$. This contradicts either the finiteness of N or $N \subseteq \operatorname{int}(B)$ and hence, L is no local lower bound set. To prove this for U one could choose $\bar{y} = (1,1,\bar{y}_3)^\top$, $\bar{y}_3 \in (z_3, Z_3)$ and use the same arguments.

Such sets $L, U \subseteq \mathbb{R}^m$ for which it is not known whether they are local lower and local upper bound sets can also appear in practice, for example using the HyPaD algorithm, see [14]. This algorithm decomposes a multi-objective mixed-integer convex optimization problem into several multi-objective continuous convex subproblems by fixing the integer assignments to certain values. Then it computes lower bound sets for the nondominated sets of those subproblems and finally merges them to an overall lower bound set of the nondominated set of the original mixed-integer problem. For this final lower bound set $L \subseteq \mathbb{R}^m$ it is not guaranteed that there exists some $N^2 \subseteq \operatorname{int}(B)$ with $L = L(N^2)$.
Thus, another strategy is needed to handle such situations and this is one of the reasons to introduce a generalization of the local upper and local lower bound concepts from Definitions 3.1 and 3.2.

4 Generalized local upper and local lower bounds

In this section we introduce a generalization of the concepts from Definitions 3.1 and 3.2. This will allow us to initialize the computation of an enclosure with an arbitrary choice of finite sets $L, U \subseteq \mathbb{R}^m$ as in Definition 2.3. In particular, this includes such settings as in Example 3.4. What is more, the generalized concepts of local upper and local lower bounds also allow for algorithmic improvements in the computation of an enclosure. We will discuss this in more detail in Section 5 where we present our new algorithm.
So far, in Section 3, we assumed the set B to be a single box $B = [z,Z]$ with $z, Z \in \mathbb{R}^m$ and $f(S) \subseteq \operatorname{int}(B)$. This assumption guarantees that for the nondominated set $\mathcal{N}$ of (MOP) and for $N = \emptyset$ it holds that $\mathcal{N} \subseteq f(S) \subseteq \operatorname{int}(B) = s(N) = S(N)$. However, there also exist other choices of $B \subseteq \mathbb{R}^m$ with $\mathcal{N} \subseteq \operatorname{int}(B)$. Moreover, while our aim is to compute an enclosure of the nondominated set, there is no need to directly include this in the definition of local upper and local lower bounds. In fact, search regions can be defined independently of what we search for. Thus, for the forthcoming Definitions 4.1 and 4.2, we denote by $B \subseteq \mathbb{R}^m$ an arbitrary area of interest with $\operatorname{int}(B) \neq \emptyset$. In particular, we no longer assume that B is a box. Further, we omit the restriction $N \subseteq \operatorname{int}(B)$ from Definitions 3.1 and 3.2 and allow $N \subseteq \mathbb{R}^m$. This has some advantages when it comes to implementational details (see also

Section 5) and allows to potentially eliminate empty boxes within an enclosure, see the forthcoming Example 5.1. Further, we replace the condition (ii) from Definitions 3.1 and 3.2 by a slightly weaker formulation. This is mainly because we need to allow empty search zones for the theoretical results in this section, see for example the forthcoming Lemma 4.4. We provide more details on the algorithmic advances of that modification on page 15. With these three changes, we obtain the following generalized definitions of local upper and local lower bounds.

Definition 4.1 *Let $N \subseteq \mathbb{R}^m$ be a finite and stable set. Then the lower search region for N is $s(N) := \{y \in \operatorname{int}(B) \mid y' \not\leq y \text{ for every } y' \in N\}$ and the lower search zone for some $u \in \mathbb{R}^m$ is $c(u) := \{y \in \operatorname{int}(B) \mid y < u\}$. A set $U = U(N) \subseteq \mathbb{R}^m$ is called local upper bound set given N if*

(i) $s(N) = \bigcup_{u \in U(N)} c(u)$,

(ii) $\{u^1\} - \operatorname{int}(\mathbb{R}^m_+) \not\subseteq \{u^2\} - \operatorname{int}(\mathbb{R}^m_+)$ *for all* $u^1, u^2 \in U(N)$, $u^1 \neq u^2$.

Each point $u \in U(N)$ is called a local upper bound (LUB).

Definition 4.2 *Let $N \subseteq \mathbb{R}^m$ be a finite and stable set. Then the upper search region for N is $S(N) := \{y \in \operatorname{int}(B) \mid y' \not\geq y \text{ for every } y' \in N\}$ and the upper search zone for some $l \in \mathbb{R}^m$ is $C(l) := \{y \in \operatorname{int}(B) \mid y > l\}$. A set $L = L(N) \subseteq \mathbb{R}^m$ is called local lower bound set given N if*

(i) $S(N) = \bigcup_{l \in L(N)} C(l)$,

(ii) $\{l^1\} + \operatorname{int}(\mathbb{R}^m_+) \not\subseteq l^2 + \operatorname{int}(\mathbb{R}^m_+)$ *for all* $l^1, l^2 \in L(N)$, $l^1 \neq l^2$.

Each point $l \in L(N)$ is called a local lower bound (LLB).

Now, we can use Definitions 4.1 and 4.2 in order to realize an initialization of an enclosure with arbitrary lower and upper bound sets as in Example 3.4. This means that, for instance, we could choose two finite and stable sets of lower and upper bounds $L', U' \subseteq \mathbb{R}^m$ as in Definition 2.3 and use $B := \mathcal{A}(L', U') = (L' + \mathbb{R}^m_+) \cap (U' - \mathbb{R}^m_+)$ as our corresponding area of interest. If we additionally ensure that $\mathcal{N} \subseteq \operatorname{int}(B)$, we obtain from Definitions 4.1 and 4.2 that the assigments of $U(\emptyset) := U'$ and $L(\emptyset) := L'$ yield indeed a local upper bound and a local lower bound set. The corresponding initialization of the enclosure is then obtained as $\mathcal{A}(L(\emptyset), U(\emptyset)) = \mathcal{A}(L', U')$. Thus, an arbitrary enclosure can be used to initialize an enclosure based on local upper and local lower bound sets in the sense of Definitions 4.1 and 4.2. For the remaining part of this paper we always assume $B \subseteq \mathbb{R}^m$ to be such a set as defined above.

Assumption 4.3 *The initial enclosure $B \subseteq \mathbb{R}^m$ is given as*

$$B := \mathcal{A}(L', U') = (L' + \mathbb{R}^m_+) \cap (U' - \mathbb{R}^m_+)$$

where $L', U' \subseteq \mathbb{R}^m$ denote two finite and stable sets of lower and upper bounds as in Definition 2.3. Further, it holds that $\mathcal{N} \subseteq \operatorname{int}(B)$.

We want to point out that the assumption $\mathcal{N} \subseteq \operatorname{int}(B)$ is needed to assure that the nondominated set is contained in the lower and upper search regions for $N = \emptyset$. An enclosure $\mathcal{A}(L', U') =: B$ in general only assures $\mathcal{N} \subseteq B$, see also the bound sets from Example 3.4. However, by replacing L' with $L' - \{\sigma e\}$ and U' by $U' + \{\sigma e\}$, where

$\sigma > 0$ denotes a small offset and $e \in \mathbb{R}^m$ denotes the all-ones vector, this gap can easily be closed. We remark that in comparison to the original assumption from Section 3, which was $f(S) \subseteq \operatorname{int}(B)$, the new Assumption 4.3 allows to choose a much tighter initialization.

Further, we want to mention that the local upper and local lower bound sets from Definitions 4.1 and 4.2 are not necessarily unique or finite. However, for $B \subseteq \mathbb{R}^m$ as in Assumption 4.3 and any finite and stable set $N \subseteq \mathbb{R}^m$ there exist a finite local upper bound set $U(N)$ and a finite local lower bound set $L(N)$. In particular, the lower and upper bound sets computed by Algorithms 1 and 2, which are still applicable as they are independent of B, are finite, see the forthcoming Lemma 4.6.

When replacing the local upper and local lower bound concepts from Definitions 3.1 and 3.2 by those from Definitions 4.1 and 4.2, we cannot rely on the theoretical results from [21] and [13, 14] any longer. This also holds regarding the update procedures, i.e., Algorithms 1 and 2, whose correctness is a key ingredient for the correctness of most enclosure algorithms, including [11, 12, 13, 14] as well as our forthcoming Algorithm 3 (AdEnA) in Section 5.

Thus, in the next part of this section we show that all theoretical results from [21] and [13, 14] that are needed in order to compute an enclosure are still valid using Definitions 4.1 and 4.2 and a choice of $B \subseteq \mathbb{R}^m$ as in Assumption 4.3. The following lemma is crucial to ensure that Algorithms 1 and 2 still compute valid local upper bound and local lower bound sets. Due to symmetry, we consider here only the result for local upper bounds.

Lemma 4.4 *Let $N \subseteq \mathbb{R}^m$ together with some $U(N)$ satisfying Definition 4.1 be given. Further, let $u \in U(N)$ and $\bar{y} \in \mathbb{R}^m$ with $\bar{y} < u$. Then it holds*

$$c(u) \setminus \{y \in s(N) \mid \bar{y} \leq y\} = \bigcup_{j=1}^{m} c(\bar{y}_j, u_{-j}).$$

Proof. Using Definition 4.1 the set $c(u) \setminus \{y \in s(N) \mid \bar{y} \leq y\}$ can be rewritten as

$$\{y \in \operatorname{int}(B) \mid y < u\} \setminus \left\{ y \in \bigcup_{u' \in U(N)} c(u') \;\middle|\; \bar{y} \leq y \right\}.$$

Using the definition of the lower search zone this equals

$$\{y \in \operatorname{int}(B) \mid y < u\} \setminus \{y \in \operatorname{int}(B) \mid \exists u' \in U(N) \colon y < u', \bar{y} \leq y\}. \tag{4.1}$$

Both sets now share the same base set $\operatorname{int}(B)$ and hence we can reformulate (4.1) as

$$\{y \in \operatorname{int}(B) \mid y < u \wedge ((\forall u' \in U(N) : y \not< u') \vee (\bar{y} \not\leq y))\}.$$

This finally reduces to

$$\{y \in \operatorname{int}(B) \mid y < u, \bar{y} \not\leq y\} = \{y \in \operatorname{int}(B) \mid \exists j \in [m] \colon y < (\bar{y}_j, u_{-j})\}$$

which is the same as $\bigcup_{j=1}^{m} c(\bar{y}_j, u_{-j})$. □

Based on the same steps as in the proofs of [21, Proposition 3.1 and Proposition 3.2] with Lemma 4.4 as a key ingredient, the following generalization of the correctness results from [21] holds.

Lemma 4.5 *Let $\emptyset \neq N \subseteq \mathbb{R}^m$ be a finite and stable set and $B \subseteq \mathbb{R}^m$ as in Assumption 4.3. Starting with $U(\emptyset) = U'$ and applying Algorithm 1 iteratively on the points of N returns a local upper bound set $U(N)$ in the sense of Definition 4.1.*

Due to symmetry, the same holds for Algorithm 2 and the computation of local lower bound sets. Hence, starting with $U(\emptyset) = U'$ and $L(\emptyset) = L'$, where $U', L' \in \mathbb{R}^m$ denote the initial upper and lower bound sets from Assumption 4.3, and then applying Algorithms 1 and 2 we obtain valid local upper bound and local lower bound sets in the sense of Definitions 4.1 and 4.2. As a prerequisite for the generalization of Lemma 3.3, we need to show that for a finite and stable set $N \subseteq \mathbb{R}^m$ there exists a finite local upper bound set $U(N)$. Due to symmetry, this result naturally extends to the local lower bound set $L(N)$ as well.

Lemma 4.6 *Let $N \subseteq \mathbb{R}^m$ be a finite and stable set and $B \subseteq \mathbb{R}^m$ as in Assumption 4.3. Then the local upper bound set $U(N)$ obtained by initializing $U(\emptyset) = U'$ and applying Algorithm 1 iteratively on the points of N is finite.*

Proof. Since U' is finite, the initial local upper bound set $U(\emptyset) = U'$ is finite. Now let $N \subseteq \mathbb{R}^m$ be a finite and stable set. By Lemma 4.5 we know that iteratively applying Algorithm 1 using the update points $y \in N$ returns a local upper bound set $U(N)$. Let $k \in \mathbb{N}$ be the number of local upper bounds before applying Algorithm 1 for the first time, i.e., $k = |U'|$. Then, using the notation from Algorithm 1, we have that $|A| \leq k$ and $|P_i| \leq mk$ for all $i \in [m]$ when using Algorithm 1 for the first time. Thus, the size of the local upper bound set after applying Algorithm 1 once is bounded by $k + m(mk)$, i.e., $|U(\{y\})| \leq k(m^2 + 1)$. Since N is a finite set this implies that also $U(N)$ is finite. □

The forthcoming lemma is a generalization of Lemma 3.3 and is a key ingredient for the correctness of the computation of an enclosure using such algorithms as those from [13, 14] or our new algorithm AdEnA, see Section 5. The proof is in large parts identical to those from [13] and [14]. However, having a different initial enclosure B as given in Assumption 4.3 and waiving the condition $N^1, N^2 \subseteq \operatorname{int}(B)$ involves some minor changes. For this reason, we give the full proof.

Lemma 4.7 *Let $B \subseteq \mathbb{R}^m$ as in Assumption 4.3 and let $N^1 \subseteq f(S) + \mathbb{R}^m_+$ and $N^2 \subseteq \mathbb{R}^m \setminus (f(S) + \operatorname{int}(\mathbb{R}^m_+))$ be finite and stable. Further, let $U(N^1)$ and $L(N^2)$ be some finite local upper and local lower bound sets in the sense of Definitions 4.1 and 4.2. Then $U(N^1)$ is an upper bound set and $L(N^2)$ is a lower bound set in the sense of Definition 2.3.*

Proof. Let $N^1 \subseteq f(S) + \mathbb{R}^m_+$ be a finite and stable set. First, we show that $\mathcal{N} \subseteq \operatorname{cl}(s(N^1))$. So let $\bar{y} \in \mathcal{N} \subseteq \operatorname{int}(B)$ be a nondominated point. Assume that $\bar{y} \notin s(N^1)$. Then, by Definition 4.1, there exists $y' \in N^1 \subseteq f(S) + \mathbb{R}^m_+$ with $y' \leq \bar{y}$. Since $\bar{y}$ is nondominated, this implies that $y' = \bar{y}$. By our assumptions it is $\bar{y} \in \operatorname{int}(B)$ and hence, there exists $\varepsilon > 0$ such that $\bar{y} - \varepsilon e \in \operatorname{int}(B)$. Hence, for $y^k := \bar{y} - \frac{\varepsilon}{k} e$, $k \in \mathbb{N}$ it holds that $(y^k)_{k\in\mathbb{N}} \subseteq \operatorname{int}(B)$. Since $y^k < y'$ for all $k \in \mathbb{N}$ and N^1 is stable, it also holds $(y^k)_{k\in\mathbb{N}} \subseteq s(N^1)$. Thus, $\bar{y} := \lim_{k\to\infty} y^k \in \operatorname{cl}(s(N^1))$ and $\mathcal{N} \subseteq \operatorname{cl}(s(N^1))$. Since, $U(N^1)$

is finite, we obtain that

$$\mathcal{N} \subseteq \operatorname{cl}(s(N^1)) = \operatorname{cl}\left(\bigcup_{u \in U(N^1)} c(u)\right) = \bigcup_{u \in U(N^1)} \operatorname{cl}(c(u)) \subseteq \bigcup_{u \in U(N^1)} \{u\} - \mathbb{R}^m_+ = U(N^1) - \mathbb{R}^m_+.$$

Hence, $U(N^1)$ is an upper bound set in the sense of Definition 2.3.
Next, we consider the local lower bound set and follow basically the same idea. Let $N^2 \subseteq \mathbb{R}^m \setminus (f(S) + \operatorname{int}(\mathbb{R}^m_+))$ be a finite and stable set. Analogously to the upper bound scenario, we first show that it holds $\mathcal{N} \subseteq \operatorname{cl}(S(N^2))$. Let $\bar{y} \in \mathcal{N} \subseteq \operatorname{int}(B)$ be a nondominated point. Assume that $\bar{y} \notin S(N^2)$. Since $\bar{y} \in \mathcal{N} \subseteq \operatorname{int}(B)$ there exists $\varepsilon > 0$ such that $\bar{y} + \varepsilon e \in \operatorname{int}(B)$. Again, we define a sequence $(y^k)_{k \in \mathbb{N}}$ with $y^k := \bar{y} + \frac{\varepsilon}{k} e \in \operatorname{int}(B)$, $k \in \mathbb{N}$. For all $k \in \mathbb{N}$ it holds that $y^k \in f(S) + \operatorname{int}(\mathbb{R}^m_+)$ which implies $(y^k)_{k \in \mathbb{N}} \subseteq S(N^2)$ by our assumptions for N^2 and Definition 4.2. As a result, $\bar{y} = \lim_{k \to \infty} y^k \in \operatorname{cl}(S(N^2))$. Finally, by Definition 4.2 and since $L(N^2)$ is finite, we obtain that

$$\mathcal{N} \subseteq \operatorname{cl}(S(N^2)) = \operatorname{cl}\left(\bigcup_{l \in L(N^2)} C(l)\right) = \bigcup_{l \in L(N^2)} \operatorname{cl}(C(l)) \subseteq \bigcup_{l \in L(N^2)} \{l\} + \mathbb{R}^m_+ = L(N^2) + \mathbb{R}^m_+.$$

□

We want to point out that all of the above results are not depending on the use of a particular algorithm to compute an enclosure of the nondominated set $\mathcal{N}$ of (MOP). They only make use of the definitions of an enclosure, local upper bounds, and local lower bounds, i.e., Definitions 2.3, 4.1 and 4.2. In particular, any algorithm that uses the concept of local upper bounds and/or local lower bounds for the computation of the bound sets L and U from Definition 2.3 can potentially make use of the presented warm start strategies.

5 Advanced Enclosure Algorithm (AdEnA)

In this section, we present our new algorithm to compute an enclosure of the nondominated set of (MOP), which we refer to as Advanced Enclosure Algorithm (AdEnA). It combines the mechanisms used to handle the patch subproblems in HyPaD, see [14, Section 4], with the generalized local upper and local lower bound techniques from the previous section. This makes it a vastly improved and generalized version of BAMOP from [13]. In particular, BAMOP and our new algorithm AdEnA can solve the exact same class of optimization problems. First, we provide a brief description of the algorithm, followed by its pseudocode and finally the theoretical results proving its correctness.
To improve the initial enclosure, see Assumption 4.3, we need to compute update points satisfying the assumptions from Lemma 4.7. Of course, we aim to compute such update points that lead to a particularly large improvement of our quality measure, i.e., the width $w(\mathcal{A})$. Since the width of an enclosure is given as the maximal shortest

edge length $s(l,u)$ of one of its boxes $[l,u]$, it would be a natural approach to try to improve that box first. However, the bound sets L and U typically grow very large and hence identifying such a largest box is often computationally costly. Thus, we compute update points using a slightly different strategy. We start by looping through all current lower bounds $l \in L$. Since it holds $\mathcal{N} \subseteq L + \mathbb{R}^m_+$ this ensures that our search for new update points covers the whole nondominated set. Then, for each lower bound $l \in L$ we search for the upper bound $u \in U$ with maximal $s(l,u)$. This selection criterion exactly matches our motivation to obtain the largest possible improvement of the width of the enclosure. We then perform the search for update points by solving

$$\min_{x,t} \; t \quad \text{s.t.} \quad f(x) - l - t(u-l) \leq 0_m, \quad x \in S, \; t \in \mathbb{R}. \tag{SUP(l,u)}$$

It is shown in [18, Proposition 2.3.4 and Theorem 2.3.1] that for all $l, u \in \mathbb{R}^m$ with $l < u$, which is exactly the setting within our algorithm, there exists an optimal solution $(\bar{x}, \bar{t})$ of (SUP(l,u)). Further, for every optimal solution $(\bar{x}, \bar{t})$ of (SUP(l,u)) the corresponding image point $f(\bar{x}) \in f(S)$ is a weakly nondominated point of (MOP), see [26, Theorem 3.2], and $l + \bar{t}(u-l) \notin f(S) + \operatorname{int}(\mathbb{R}^m_+)$. Hence, by Lemma 4.7 these points can be used as update points for the local upper and local lower bound sets L and U as intended. The procedure described above is repeated until $w(\mathcal{A}(L,U)) \leq \varepsilon$, which leads to our Advanced Enclosure Algorithm (AdEnA) as presented in Algorithm 3.

Algorithm 3 Advanced Enclosure Algorithm (AdEnA)

Input: Quality parameter $\varepsilon > 0$, initial bound sets L', U' as in Assumption 4.3
Output: Sets L, U of lower and upper bounds of $\mathcal{A}(L,U)$

procedure AdEnA(ε, L', U')
  Initialize $L = L', U = U'$
  while $w(\mathcal{A}(L,U)) > \varepsilon$ **do**
    for $l \in L$ **do**
      if $(\{l + \varepsilon e\} + \operatorname{int}(\mathbb{R}^m_+)) \cap U \neq \emptyset$ **then**
        Select $u \in (\{l + \varepsilon e\} + \operatorname{int}(\mathbb{R}^m_+)) \cap U$ with maximal $s(l,u)$
        Solve (SUP(l,u)) with optimal solution $(\bar{x}, \bar{t})$ and set
          $\bar{y} := f(\bar{x})$ and $\hat{y} := l + \bar{t}(u-l)$
        $L =$ UpdateLLB$(L, \hat{y})$
        $U =$ UpdateLUB$(U, \bar{y})$
      end if
    end for
  end while
end procedure

The following example illustrates a single update step of Algorithm 3 and motivates why we allow update points outside of $\operatorname{int}(B)$ in Definitions 4.1 and 4.2.

Example 5.1 We consider the multi-objective mixed-integer convex quadratic optimization problem

$$\begin{aligned} \min_{x} \; & (x_1 + x_3, x_2 + x_4)^\top \quad \text{s.t.} \quad x_1^2 + x_2^2 \leq 0.25, \\ & x_3^2 + x_4^2 \leq 1, \\ & x_1, x_2 \in \mathbb{R}, \\ & x_3, x_4 \in \mathbb{Z}. \end{aligned} \tag{Ex2}$$

Further, we choose $L' := \{l^1, l^2, l^3\}$ and $U' := \{u^1, u^2, u^3\}$ as shown in Figure 2. Then $B := \mathcal{A}(L', U')$ is a valid initial enclosure satisfying Assumption 4.3 with $\mathcal{N} \subseteq \operatorname{int}(B)$.

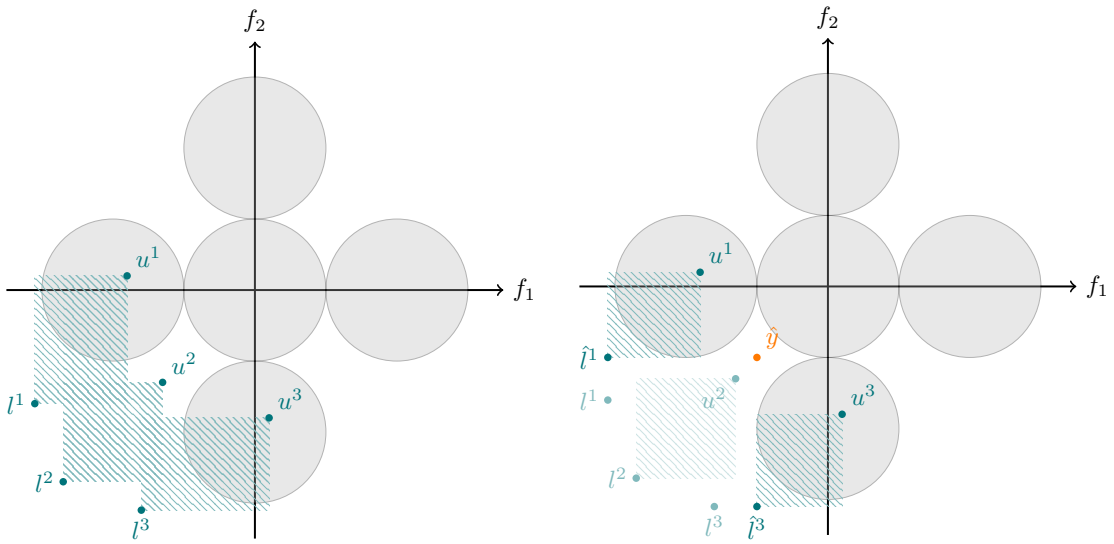

Figure 2: Enclosure of the nondominated set of (Ex2) before and after updating the lower bound set by $\hat{y} \notin \operatorname{int}(B)$

Next, we discuss only the update of the lower bound set. Let $l = l^2$, $u = u^2$ and denote by $(\bar{x}, \bar{t})$ an optimal solution of (SUP(l, u)). Then the lower bound set $L = L'$ is updated by $\hat{y} := l + \bar{t}(u - l)$. As a result, the empty box $[l^2, u^2]$ is removed from the enclosure, where empty means that $\mathcal{N} \cap [l^2, u^2] = \emptyset$.

The removement of the (empty) box $[l^2, u^2]$ in Example 5.1 is only possible because Definition 4.2 allows to use update points $\hat{y} \notin \operatorname{int}(B)$. This is not the case for Definition 3.2. Nevertheless, such update points $\hat{y} \notin \operatorname{int}(B)$ can also appear within enclosure algorithms from the literature that make use of the classic local lower bound concept. This is the reason why in the literature the update procedures for the lower bound set in such a setting require to use some kind of workaround. For instance, in [14, Algorithm 4] this workaround is realized by using projections of such points $\hat{y}$ onto $\operatorname{int}(B)$. Hence, the advantage of the generalized Definitions 4.1 and 4.2 is not only to make the update of the bound sets easier, but also to potentially eliminate empty boxes from the enclosure.
We want to add a brief note on empty search zones. Besides the fact that allowing empty search zones was necessary for the theoretical results in the previous section, especially for Lemma 4.4, they keep Algorithm 3 simple. In particular, there is no need to filter out empty search zones within the Algorithms 1 and 2, which can become computationally costly. What is more, bounds that correspond to empty search zones implicitly become inactive anyway. Let $l \in L$ be a local lower bound with $C(l) = \emptyset$. Then the condition in line 5 of Algorithm 3 is never satisfied. Hence, such local lower bounds can be considered inactive. The same holds for local upper bounds. Let $u \in U$ with $c(u) = \emptyset$. Then there exists no $l \in L$ with $u \in (\{l + \varepsilon e\} + \operatorname{int}(\mathbb{R}^m_+))$. Thus, such local upper bounds are never chosen in line 6 of Algorithm 3 and can also be considered inactive. So especially from an algorithmic point of view there are no drawbacks of allowing empty search zones.

Moreover, we note that, while all theoretical results require only continuous objective functions $f_i, i \in [m]$ and a compact, nonempty set S, for practical use of AdEnA one needs to be able to solve (SUP(l, u)) fast and reliably. Due to ongoing advances in research and solver development, the class of optimization problems for which this is possible is constantly growing. It includes not only continuous convex optimization problems, but for example also mixed-integer linear and mixed-integer quadratic optimization problems. We present numerical results for these problem classes later in Section 6.

Since our new algorithm AdEnA can solve the same class of optimization problems as BAMOP, see [13, Algorithm 3], next we point out the main differences between both algorithms before we move on to the proof of correctness for AdEnA. First of all, BAMOP uses the classic local upper and local lower bound concepts from Section 3. In its original form, it only allows to use a single box $B := [z, Z]$ for the initialization of the enclosure. Besides that, warm starts as presented in Section 3 would also be possible. However, an initialization with arbitrary lower and upper bound sets as in Assumption 4.3 is not possible for BAMOP with those classic concepts.

The main drawbacks of BAMOP result from its procedure to update the lower and upper bound sets. While AdEnA only selects one upper bound $u \in U$ for each lower bound $l \in L$, see line 6 in Algorithm 3, BAMOP loops through all the upper bounds $u \in (\{l + \varepsilon e\} + \operatorname{int}(\mathbb{R}^m_+)) \cap U$. Since all boxes $[l, u]$ within such a loop share the same lower bound $l \in L$, this might restrict the improvements of the overall enclosure to a small area of the image space. What is more, since the upper bound set for this loop is fixed, but updated within the loop, there might be a lot of boxes that have already been improved when the first update point in the loop was computed. Thus, there might be a lot of update points that actually do not result in a further improvement of the upper bound set, especially for the later iterations of the corresponding for loop. This drawback is already shortly discussed in [13, Section 5.2] and also the use of a selection criterion as in AdEnA is suggested. However, it is not clear whether the theoretical results of BAMOP, especially concerning its correctness and finiteness, still hold when working with a selection criterion. For AdEnA, this gap is closed by the forthcoming Theorem 5.2.

A further disadvantage of BAMOP is the computation of the update points themselves. BAMOP relies on nondominated points $\bar{y} \in f(S)$ of (MOP) for the update points. Just weakly nondominated points are not sufficient. For this reason, after having solved (SUP(l, u)) for a box $[l, u]$, another optimization problem is solved. The latter computes a nondominated point which dominates the given weakly nondominated point, see [13, Section 4.2].

In addition to that, BAMOP relies on a special proof of correctness, mainly the Halving Theorem ([13, Theorem 4.2]). This requires a more complex computation of update points for the lower bounds. While this involves basically just to check whether two points $y^1, y^2 \in \mathbb{R}^m$ are identical, even this little computational effort sums up over time. A comparison of computation times for AdEnA and BAMOP is provided in Section 6 for selected test instances.

For the remaining part of this section we focus on the theoretical results for Algorithm 3. The proof of the following theorem is similar to the proof of [14, Theorem 4.1]. The differences come from the fact that Algorithm 3 makes use of the new theoretical foundation based on the generalized local upper and local lower bounds from the previous section. It shows that with each iteration, i.e., with each run of the while

loop, Algorithm 3 ensures an improvement of the enclosure $\mathcal{A} = \mathcal{A}(L, U)$ by a certain amount. We remark that the considered volume $\mathrm{vol}(\mathcal{A})$ is easy to evaluate since an enclosure is just a combination of boxes.

Theorem 5.2 *Let $L', U' \subseteq \mathbb{R}^m$ as in Assumption 4.3 with $\mathcal{N} \subseteq \mathrm{int}(\mathcal{A}(L', U'))$ and $\varepsilon > 0$ be the input parameters for Algorithm 3. Denote by L^{start} and U^{start} the lower and upper bound sets at the beginning of some iteration of Algorithm 3 and by L^{end}, U^{end} the sets at the end of the same iteration. Additionally, assume that $w(\mathcal{A}(L^{start}, U^{start})) > \varepsilon$. Then, within that iteration, the volume of the enclosure $\mathcal{A}$ is reduced by at least $(\varepsilon/2)^m$, i.e.,*

$$\mathrm{vol}(\mathcal{A}(L^{end}, U^{end})) < \mathrm{vol}(\mathcal{A}(L^{start}, U^{start})) - \left(\frac{\varepsilon}{2}\right)^m.$$

Proof. Since $w(\mathcal{A}(L^{\mathrm{start}}, U^{\mathrm{start}})) > \varepsilon$, there exist $l \in L^{\mathrm{start}}, u \in U^{\mathrm{start}}$ such that $u \in (\{l + \varepsilon e\} + \mathrm{int}(\mathbb{R}^m_+))$, i.e., $u_i - l_i > \varepsilon$ for all $i \in [m]$. We can assume w.l.o.g. that these bounds correspond exactly to the assignment of l and u when line 6 of Algorithm 3 is reached for the first time within the current iteration. In the following, we use the notation from Algorithm 3.
First, we consider the case $\bar{t} > 0.5$. Denote by $\tilde{t} := \min\{\bar{t}, 1\} > 0.5$ and set $\tilde{y} := l + \tilde{t}(u - l) > 0.5(u + l)$. We update the lower bound set L using Algorithm 2 with update point $\hat{y} := l + \bar{t}(u - l)$, see line 8 of Algorithm 3. By Definition 4.2, this implies that the set $\{y \in \mathrm{int}(B) \mid y \leq \hat{y}\}$ is removed from the upper search region. In particular, the open box $(l, \tilde{y}) \subseteq \{y \in \mathrm{int}(B) \mid y \leq \hat{y}\} \cap \mathcal{A}$ is removed from the upper search region and from the enclosure $\mathcal{A}$. Since we have that

$$\mathrm{vol}((l, \tilde{y})) = \prod_{i=1}^{m} (\tilde{y}_i - l_i) > \prod_{i=1}^{m} (0.5(u_i + l_i) - l_i) > \prod_{i=1}^{m} 0.5\,\varepsilon = \left(\frac{\varepsilon}{2}\right)^m,$$

we obtain that $\mathrm{vol}(\mathcal{A}(L^{\mathrm{end}}, U^{\mathrm{end}})) < \mathrm{vol}(\mathcal{A}(L^{\mathrm{start}}, U^{\mathrm{start}})) - (\varepsilon/2)^m$.
The second case to consider is $\bar{t} \in [0, 0.5]$. In that case, the upper bound set U is updated using Algorithm 1 with update point $f(\bar{x}) \leq l + \bar{t}(u - l) \leq 0.5(u + l)$. By Definition 4.1 this implies that the open box $(f(\bar{x}), u)$ is removed from the lower search region and in particular from the enclosure $\mathcal{A}$. Since it holds that

$$\mathrm{vol}((f(\bar{x}), u)) = \prod_{i=1}^{m} (u_i - f_i(\bar{x})) \geq \prod_{i=1}^{m} (u_i - 0.5(u_i + l_i) > \prod_{i=1}^{m} 0.5\,\varepsilon = \left(\frac{\varepsilon}{2}\right)^m,$$

this implies that $\mathrm{vol}(\mathcal{A}(L^{\mathrm{end}}, U^{\mathrm{end}})) < \mathrm{vol}(\mathcal{A}(L^{\mathrm{start}}, U^{\mathrm{start}})) - (\varepsilon/2)^m$.
Finally, we remark that the case $\bar{t} < 0$ cannot occur. Otherwise, if $\bar{t} < 0$ there exists $\bar{x} \in S$ with $\bar{y} = f(\bar{x}) < l$. On the other hand, by Lemma 4.7 there exists $l' \in L^{\mathrm{start}}$ with $l' \leq f(\bar{x})$. Hence, there exist $l, l' \in L^{\mathrm{start}}$ such that $l' < l$, which contradicts Definition 4.2. □

Since the volume $\mathrm{vol}(\mathcal{A}(L', U'))$ of the initial enclosure is finite and reduced by at least $(\varepsilon/2)^m$ with each iteration of the while loop according to Theorem 5.2, the volume decreases to zero within a finite number of iterations. Since for all boxes with a volume less than ε^m also the shortest edge length is less than ε, this immediately implies finiteness and correctness of Algorithm 3.

Corollary 5.3 *Algorithm 3 is finite, i.e., after a finite number of iterations it holds $w(\mathcal{A}(L, U)) \leq \varepsilon$.*

Finally, we remark that, as a byproduct, Algorithm 3 also computes a finite representation of the weakly nondominated set of (MOP). This representation is given by the set of points $\bar{y} \in f(S)$ computed in line 7 of the algorithm.

6 Numerical Results

In this section we present our numerical results for selected test instances. Those test instances cover various classes of optimization problems for which a fast and reliable solver for the subproblems (SUP(l,u)) is available. More precisely, we present continuous convex, mixed-integer convex quadratic and mixed-integer non-convex quadratic examples. All numerical tests have been performed using MATLAB R2021a on a machine with Intel Core i9-10920X processor and 32GB of RAM. The average of the results of `bench(5)` is: LU = 0.2045, FFT = 0.2127, ODE = 0.3666, Sparse = 0.3919, 2-D = 0.1968, 3-D = 0.2290. Be aware that, according to the MATLAB documentation, these values are version specific, see [23].
In order to provide a fair comparison of our algorithm AdEnA (see Algorithm 3) with BAMOP [13] and HyPaD [14], we make use of the same subsolvers and parameters wherever applicable. This means that all single-objective mixed-integer optimization problems within our algorithm are solved using GUROBI [17] and all single-objective (purely continuous) convex optimization problems are solved using `fmincon`. For most instances, we set $\varepsilon = 0.1$ as termination criterion for all algorithms. If not stated otherwise, we use a single box $B := [z, Z] \subseteq \mathbb{R}^m$ to initialize the enclosure, i.e., $L' = \{z\}$, $U' = \{Z\}$, to allow for a fair comparison between our algorithm and HyPaD or BAMOP. Of course, we also discuss other initializations using warm start strategies as suggested in Section 4. Besides that all parameters are chosen as in the original papers [13] and [14]. For all test instances we used a time limit of 3600 seconds.

Test instance 1 First, we consider a bi-objective mixed-integer convex quadratic test instance from [3]. It is scalable in the number $k \in \mathbb{N}$ of continuous and $l \in \mathbb{N}$ of integer variables, where k needs to be even and the total number of variables is $n = k + l$.

$$\min_x \left(\sum_{i=1}^{k/2} x_i + \sum_{i=k+1}^{n} x_i, \sum_{i=k/2+1}^{k} x_i - \sum_{i=k+1}^{n} x_i \right)^\top \quad \text{s.t.} \quad \begin{array}{l} \sum_{i=1}^{k} x_i^2 \leq 1, \\ x \in [-2,2]^n, \\ x_i \in \mathbb{Z},\ i = k+1, \ldots, n \end{array} \tag{T4}$$

We consider 18 different combinations of the parameters $k, l \in \mathbb{N}$ as done in [14, 15] to provide a detailed comparison of AdEnA and HyPaD. Besides the results for AdEnA using the same initial box $B = [z, Z]$ that was used for HyPaD in [15], i.e., setting $L' = \{z\}$, $U' = \{Z\}$, we also include the results for a warm start strategy. For that we make use of the initial bound sets

$$L' := \left\{ \left(\sigma - \sqrt{k/2}, -\sigma - \sqrt{k/2} \right)^\top \in \mathbb{R}^2 \;\middle|\; \sigma \in \{-2l, -2l+1, \ldots, 2l\} \right\} - \{10^{-6}\, e\},$$
$$U' := \left\{ \left(\sigma + \sqrt{k/2}, -\sigma + \sqrt{k/2} \right)^\top \in \mathbb{R}^2 \;\middle|\; \sigma \in \{-2l, -2l+1, \ldots, 2l\} \right\} + \{10^{-6}\, e\}$$

which can be constructed by simple estimates. We note that the small offset of 10^{-6} is introduced to ensure that $\mathcal{N} \subseteq \operatorname{int}(\mathcal{A}(L', U'))$ holds. We refer to this configuration as

$\text{AdEnA}_{\text{warm}}$. The configuration using $L' = \{z\}$, $U' = \{Z\}$ is denoted by $\text{AdEnA}_{\text{single}}$. The results for the computation times are shown in Table 1, where '-' indicates that the algorithm did not terminate within the time limit of 3600 seconds.

k	l	$\text{AdEnA}_{\text{single}}$	$\text{AdEnA}_{\text{warm}}$	BAMOP	HyPaD
2	1	0.52	0.49	0.94	1.52
2	2	1.08	0.93	1.78	3.63
2	3	1.52	1.37	2.71	5.85
4	1	1.58	1.40	3.19	1.85
2	10	4.82	3.63	9.19	21.99
4	10	14.20	12.17	26.77	33.13
8	10	14.97	19.29	32.73	-
2	20	9.84	7.46	19.31	65.59
4	20	28.67	24.19	51.14	106.89
2	30	15.68	11.18	29.80	166.42
4	30	41.79	36.66	82.32	219.93
8	30	52.33	58.90	96.01	310.63
16	30	79.34	84.19	-	-
200	2	8.82	9.87	19.15	167.45
200	4	22.83	16.84	36.37	271.67
200	6	27.30	23.87	51.30	452.58
200	8	29.29	30.84	67.06	649.41
200	10	52.32	37.78	82.14	878.64

Table 1: Computation times in seconds for test instance (T4) using $\text{AdEnA}_{\text{single}}$, $\text{AdEnA}_{\text{warm}}$, BAMOP, and HyPaD

For all instances, both configurations of AdEnA were able to outperform BAMOP and HyPaD. In general, the advantage of AdEnA grows with the number of variables. In particular for the instances with $k = 200$, AdEnA is more than ten times faster compared to HyPaD. We also see that both configurations of AdEnA were able to compute results for all instances, whereas BAMOP was not able to do so for one and HyPaD was not able to do so for two instances within the time limit of 3600 seconds. Thus, AdEnA seems to be a promising solution approach for even larger instances of (T4). However, this is not surprising since AdEnA makes use of the quadratic structure of the problem, which leads to (single-objective) mixed-integer quadratic subproblems (SUP(l, u)). In fact, this is also the reason why BAMOP outperforms HyPaD on most instances. HyPaD, on the other hand, solves mixed-integer linear subproblems that are obtained by linearization of the objective and constraint functions. While for problems like (T4) this is a slight disadvantage, this approach allows HyPaD to solve any kind of multi-objective mixed-integer convex optimization problems. In particular, this includes such optimization problems for which fast and reliable subsolvers for (SUP(l, u)) are not (yet) available and which can for this reason not be solved using AdEnA or BAMOP. For an illustration of the results obtained by AdEnA compared to HyPaD, see Figure 3. Please note that in order to make it easier to recognize the difference in the box structure of the results, the figure shows only a section and not the whole enclosures that have been computed by the algorithms.

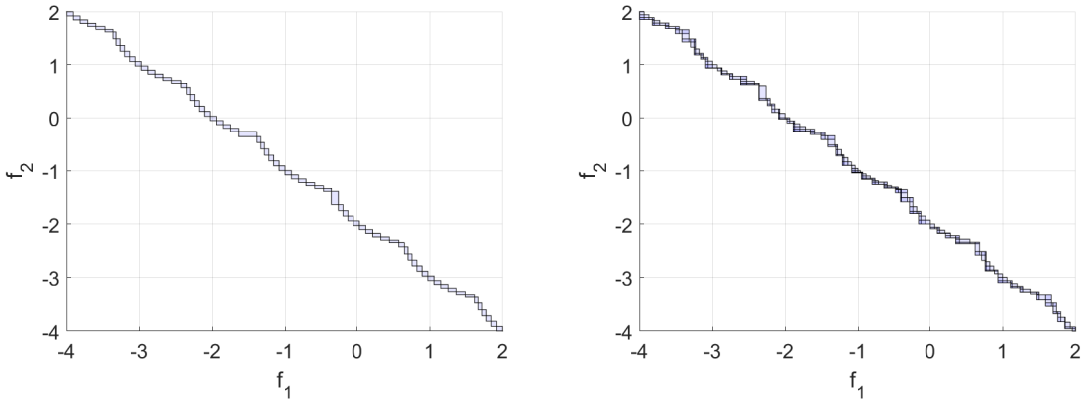

Figure 3: Comparison of the enclosures computed for (T4) with $(k, l) = (4, 10)$ by $\text{AdEnA}_{\text{single}}$ (left) and HyPaD (right); visualization restricted to $[-4, 2]^2 \subseteq f(S)$

Comparing the performance of AdEnA (more precisely $\text{AdEnA}_{\text{single}}$) and BAMOP, we observe that AdEnA is roughly twice as fast as BAMOP. This is mainly due to the fact that BAMOP needs to solve two optimization problems (SUP) and (GNP) for every update of the lower and upper bound sets, while AdEnA needs to solve only (SUP). In fact, this is exactly what Table 2 shows. In that table, we report the number of calls and the computation time for the subproblems of AdEnA and BAMOP. We remark that since the number of calls of (SUP) and (GNP) is equal for BAMOP, we reported only the number for (SUP). This means that the total number of subproblems that was solved within BAMOP equals twice the number that is listed there. Surprisingly the number of calls for (SUP) is often (almost) the same for AdEnA and BAMOP. As we will see for Test instance 3, this is in general not the case.
Comparing $\text{AdEnA}_{\text{single}}$ and $\text{AdEnA}_{\text{warm}}$, the latter has the better computation time for 13 out of the 18 test instances what demonstrates its potential as a warm start strategy. In Figure 4 we also provide a visual comparison of the results for (T4) with $(k, l) = (4, 1)$. Again, we decided to show only a section of the overall enclosures to make the differences in the composition of the enclosures easier to recognize.
However, it is rather difficult to obtain a general insight about the reduction in computation time that will be obtained by $\text{AdEnA}_{\text{warm}}$. For example, the computation times of $\text{AdEnA}_{\text{single}}$ and $\text{AdEnA}_{\text{warm}}$ are almost equal for $(k, l) = (2, 1)$ and $(k, l) = (200, 8)$. For other instances, such as $(k, l) = (2, 30)$ and $(k, l) = (200, 10)$, the computation time is decreased by more than 25 percent. But there also exist some instances where $\text{AdEnA}_{\text{warm}}$ increases the computation time. Overall, however, using $\text{AdEnA}_{\text{warm}}$ was able to reduce the computation time by roughly 12 percent (in median) compared to $\text{AdEnA}_{\text{single}}$.
We remark that, in general, additional computation time will be necessary in order to obtain an initialization to warm start AdEnA unless a good initialization is known analytically as in this example. As a result, there is always a trade-off between the time that is needed to compute a more advanced initialization and the resulting reduction in computation time for AdEnA. Thus, especially for computationally heavy problems (e.g., with a large number of variables) and whenever obtaining additional information for warm starts is in some sense computationally cheap, we recommend using $\text{AdEnA}_{\text{warm}}$ in favor of $\text{AdEnA}_{\text{single}}$.

k	l	$\text{AdEnA}_{\text{single}}$			BAMOP			
			(SUP)			(SUP)		(GNP)
		time	calls	time	time	calls	time	time
2	1	0.52	45	0.48	0.94	45	0.51	0.38
2	2	1.08	87	1.02	1.78	87	1.01	0.71
2	3	1.52	117	1.45	2.71	117	1.64	0.99
4	1	1.58	57	1.52	3.19	70	1.85	1.26
2	10	4.82	401	4.61	9.19	401	5.61	3.34
4	10	14.20	475	13.91	26.77	496	14.23	12.20
8	10	14.97	449	14.69	32.73	518	17.28	15.10
2	20	9.84	795	9.37	19.31	792	11.84	6.97
4	20	28.67	941	28.04	51.14	935	26.99	23.48
2	30	15.68	1165	14.93	29.80	1165	18.69	10.32
4	30	41.79	1367	40.81	82.32	1489	43.42	37.77
8	30	52.33	1531	51.20	96.01	1512	50.14	44.68
16	30	79.34	1837	77.98	-	-	-	-
200	2	8.82	177	8.71	19.15	195	10.39	8.62
200	4	22.83	341	22.62	36.37	312	19.88	16.28
200	6	27.30	417	27.05	51.30	428	27.96	23.05
200	8	29.29	431	29.04	67.06	523	36.50	30.21
200	10	52.32	715	51.87	82.14	631	45.34	36.38

Table 2: Number of calls and computation time in seconds for the subroutines of $\text{AdEnA}_{\text{single}}$ and BAMOP for test instance (T4)

Test instance 2 Next, we study a further multi-objective mixed-integer optimization problem. It was presented in [12] and has linear objective and nonconvex quadratic constraint functions.

$$\min_x (x_1 + x_3, x_2 + x_4)^\top \quad \text{s.t.} \quad \begin{aligned} & x_1^2 + x_2^2 \geq 1, \\ & x_3^2 + x_4^2 \leq 9, \\ & x_1, x_2 \in [0, 1], \\ & x_3, x_4 \in [-3, 3] \cap \mathbb{Z} \end{aligned} \tag{P2}$$

In particular, for this test instance, the suproblems (SUP(l, u)) are now single-objective mixed-integer nonconvex quadratic optimization problems. Due to recent advances in the development of solvers for mixed-integer optimization, e.g., considering Gurobi, we can expect that these problems can, in general, be solved fast and reliably.
Since (P2) is nonconvex, it can not be solved by HyPaD. So there are multi-objective mixed-integer optimization problems that can be solved by HyPaD, but not by AdEnA, see the discussion in the previous section, and there are optimization problems like (P2) that can be solved by AdEnA, but not by HyPaD. In particular, considering only the mixed-integer setting, AdEnA cannot be considered a simplified or adjusted version of HyPaD for some specially structured classes of objective and constraint functions.
For the initialization of AdEnA we chose $L' = \{z\}$, $U' = \{Z\}$ with $z := (-3, -3)^\top - 10^{-4}$ and $Z := (6, 6)^\top + 10^{-4}$. Within just 0.48 seconds an enclosure of the nondominated set of (P2) was obtained. For BAMOP the computation time was 0.99 seconds.

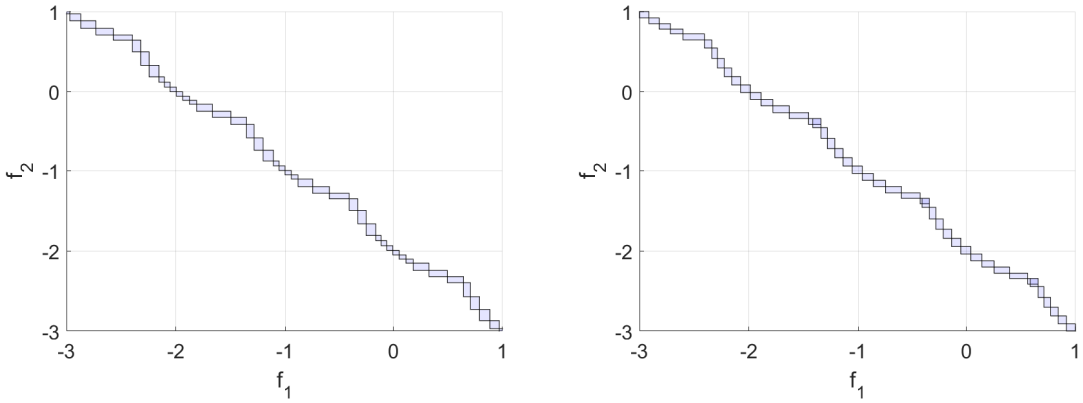

Figure 4: Comparison of the enclosures computed for (T4) with $(k, l) = (4, 1)$ by $\text{AdEnA}_{\text{single}}$ (left) and $\text{AdEnA}_{\text{warm}}$ (right); visualization restricted to $[-3, 1]^2 \subseteq f(S)$

Test instance 3 One of our motivations for the development of AdEnA was to improve BAMOP, which is suggested mainly for solving continuous multi-objective convex optimization problems, see [13]. Thus, in this final test instance we consider the following tri-objective example from [6] that was also discussed in [13].

$$\min_x \; x \quad \text{s.t.} \quad \left(\frac{x_1 - 1}{1}\right)^2 + \left(\frac{x_2 - 1}{a}\right)^2 + \left(\frac{x_3 - 1}{5}\right)^2 \leq 1, \; x \in \mathbb{R}^3 \tag{Ex5.1}$$

While the parameter $a > 0$ can be chosen arbitrarily, we consider here the exact same choices as in [13], i.e., $a \in \{5, 7, 10, 20\}$. We want to point out that since in [13] another computer setup was used for the numerical tests, we computed new results for BAMOP using the same setup as for AdEnA. This is why the computation times differ from those that are presented in [13]. To guarantee a fair comparison between AdEnA and BAMOP we chose a single box, i.e., $L' = \{z\}$, $U' = \{Z\}$, for the initialization of AdEnA. Thereby $z, Z \in \mathbb{R}^3$ are chosen as in the numerical example from [13]. The results for both, AdEnA and BAMOP, are shown in Table 3. Besides the overall computation times, we also provide the computation times for the subproblems as well as the number of calls of the subproblems. An illustration of the results for $a = 5$ is provided in Figure 5.

a	AdEnA			BAMOP				BAVS
		(SUP)			(SUP)		(GNP)	
	time	calls	time	time	calls	time	time	time
5	9.24	625	8.56	48.41	1470	18.93	28.15	17.19
7	10.68	714	9.91	57.44	1734	22.73	33.05	18.14
10	11.87	791	10.98	66.06	1995	26.06	37.99	20.62
20	14.41	926	13.39	82.98	2456	32.92	47.35	19.95

Table 3: Computation times in seconds needed to obtain an enclosure of (Ex5.1) using AdEnA and BAMOP as well as the computation time to obtain a coverage using the Benson-type algorithm with vertex selection (BAVS) from [6]

In Table 3 we also provide the computation times to obtain a coverage of the nondominated set for (Ex5.1) by the Benson-type algorithm with vertex selection (BAVS) from [6]. For that algorithm we also used the tolerance of $\varepsilon = 0.1$. However, since the coverage computed by that algorithm is not an enclosure but based on polyhedra, we focus on a brief qualitative comparison. The main difference between AdEnA and BAVS for this test instance is that the computation time of AdEnA tends to increase for larger choices of the parameter a while the computation time of BAVS stays roughly the same. This is no surprise since that is exactly one of the benefits of vertex selection, see also [6, Table 1].

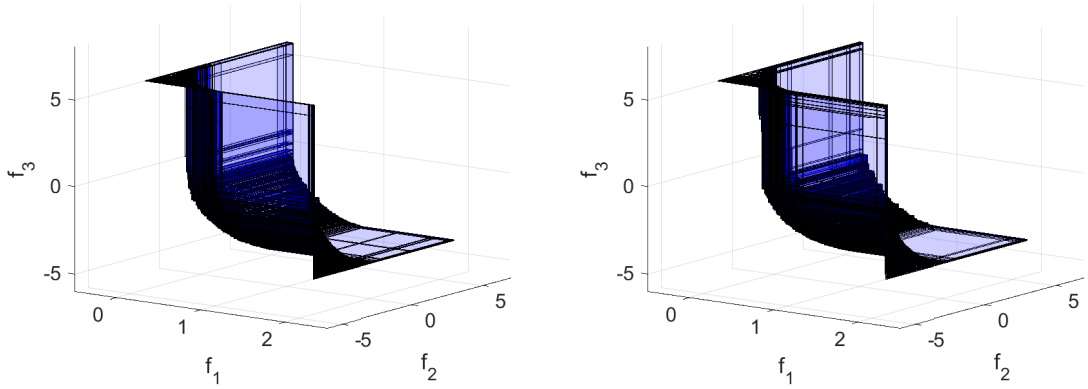

Figure 5: Comparison of the enclosures computed for (Ex5.1) with $a = 5$ by AdEnA (left) and BAMOP (right)

Comparing AdEnA and BAMOP, which both compute an enclosure of the nondominated set, AdEnA clearly outperforms BAMOP being more than five times faster for this test instance. This is exactly what we expected as a result of the improvements we incorporated in the new algorithm. In particular, compared to BAMOP, Algorithm 3 only loops through the lower bounds $l \in L$. Then it selects a single upper bound $u \in U$ to perform an update of the bound sets instead of looping through all upper bounds $u \in (\{l + \varepsilon e\} + \operatorname{int}(\mathbb{R}^m_+)) \cap U$. This is possible due to the different choice of update points for the lower bound set and the new improvement results from Theorem 5.2. Also the improvement step itself, i.e., the computation of update points for the sets L and U, requires less computational effort in AdEnA. In BAMOP two (single-objective) optimization problems need to be solved for each improvement step, while AdEnA needs to solve only one (namely (SUP(l, u))). In comparison to the results from the first test instance, see Table 2, the results in Table 3 show that the number of calls of the subproblem (SUP) can be significantly larger for BAMOP compared with AdEnA. This is exactly what one would expect due to the additional loop for the upper bounds that is an integral part of BAMOP but not necessary for AdEnA.
The idea to select a single upper bound instead of looping through a whole set of them was already briefly introduced in [13, Section 5.2]. However, there it was only a suggestion since it was not clear whether this approach results in a finite and correct algorithm. Also, replacing just the strategy for the selection of upper bounds was only able to roughly halve the computation time of BAMOP, see [13, Figure 13]. With AdEnA, it was reduced by more than 80 percent. For an illustration, see Figure 6. For $a \in \{1, 3, 5, \ldots, 25\}$ it shows the computation times of BAMOP (orange circles), AdEnA (blue circles), and also half of the computation times of BAMOP (dashed gray

line) for better orientation. Please be aware that those computation times slightly differ from the ones presented in Table 3, since we did not use the 'Run and Time' feature in that setting which normally introduces some computational overhead.

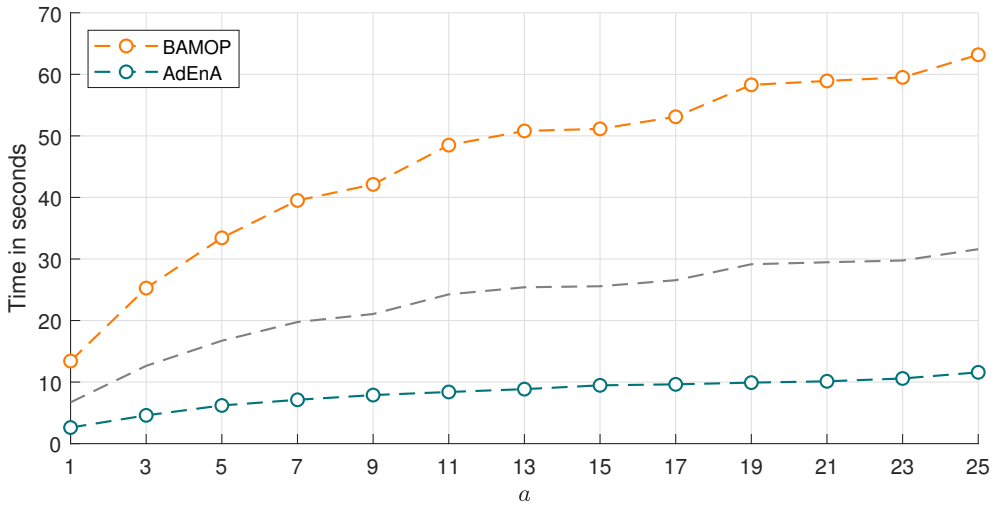

Figure 6: Comparison of the computation times of AdEnA and BAMOP for (Ex5.1)

7 Conclusions

In this paper, we presented a new algorithm AdEnA (Algorithm 3) to compute an enclosure of the nondominated set of general multi-objective optimization problems (MOP). Compared to enclosure algorithms from the literature, AdEnA can be initialized with an arbitrary enclosure $\mathcal{A}(L', U')$, see also Assumption 4.3, and not only using a single box $B = [z, Z]$. This is possible because we generalized the classic local upper bound concept from [21] and proved that the generalized bound concepts from Definitions 4.1 and 4.2 can be used to obtain lower and upper bounds as needed for an enclosure, see Lemma 4.7.
Compared to BAMOP, see [13, Algorithm 3], which is an enclosure algorithm for the same class of optimization problems (MOP), AdEnA performs up to eight times faster as we have demonstrated in Section 6. This is a result of the use of the generalized bound concepts from Section 4, but also because the update of the bound sets has been significantly simplified. The latter has mainly been possible since the improvement of the overall enclosure no longer relies on a halving property, see [13, Theorem 4.2], which allowed us to omit looping through a whole set of upper bounds and to select only a single upper bound $u \in U$ instead, see line 6 of Algorithm 3.
For all theoretical results in this paper, especially concerning the finiteness and correctness of AdEnA, we only assumed continuous objective functions f_i, $i \in [m]$ and a nonempty and compact feasible set $S \subseteq \mathbb{R}^n$. However, the practical use of the algorithm requires a fast and reliable solver for the single-objective subproblems (SUP(l, u)). Still, this makes the algorithm applicable to a large class of optimization problems as long as such a solver is available. In particular, as the development of solvers for special classes of single-objective optimization problems evolves, AdEnA can be used for more and more classes of optimization problems.

Acknowledgments

We thank the authors from [6] who generously provided us with the code of their algorithm.
This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 432218631.

References

[1] R. S. Burachik, C. Y. Kaya, and M. M. Rizvi, *Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems*, Engineering Optimization, 54 (2021), pp. 1413–1425.

[2] I. Das and J. E. Dennis, *Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems*, SIAM Journal on Optimization, 8 (1998), pp. 631–657.

[3] M. De Santis, G. Eichfelder, J. Niebling, and S. Rocktäschel, *Solving multiobjective mixed integer convex optimization problems*, SIAM Journal on Optimization, 30 (2020), pp. 3122–3145.

[4] K. Dächert, K. Klamroth, R. Lacour, and D. Vanderpooten, *Efficient computation of the search region in multi-objective optimization*, European Journal of Operational Research, 260 (2017), pp. 841–855.

[5] K. Dächert and K. Teichert, *An improved hyperboxing algorithm for calculating a Pareto front representation*. https://arxiv.org/abs/2003.14249, 2020.

[6] D. Dörfler, A. Löhne, C. Schneider, and B. Weissing, *A Benson-type algorithm for bounded convex vector optimization problems with vertex selection*, Optimization Methods and Software, 37 (2021), pp. 1006–1026.

[7] M. Ehrgott and X. Gandibleux, *Bound sets for biobjective combinatorial optimization problems*, Computers & Operations Research, 34 (2007), pp. 2674–2694.

[8] M. Ehrgott, L. Shao, and A. Schöbel, *An approximation algorithm for convex multi-objective programming problems*, Journal of Global Optimization, 50 (2010), pp. 397–416.

[9] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer, 2008.

[10] ——, *Twenty years of continuous multiobjective optimization in the twenty-first century*, EURO Journal on Computational Optimization, 9 (2021). 100014.

[11] G. Eichfelder, P. Kirst, L. Meng, and O. Stein, *A general branch-and-bound framework for continuous global multiobjective optimization*, Journal of Global Optimization, 80 (2021), pp. 195–227.

[12] G. Eichfelder, O. Stein, and L. Warnow, *A deterministic solver for multiobjective mixed-integer convex and nonconvex optimization*. http://www.optimization-online.org/DB_HTML/2022/02/8796.html, 2022.

[13] G. Eichfelder and L. Warnow, *An approximation algorithm for multi-objective optimization problems using a box-coverage*, Journal of Global Optimization, 83 (2021), pp. 329–357.

[14] ——, *A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems*. http://www.optimization-online.org/DB_HTML/2021/08/8541.html, 2021.

[15] ——, *On implementation details and numerical experiments for the HyPaD algorithm to solve multi-objective mixed-integer convex optimization problems*. http://www.optimization-online.org/DB_HTML/2021/08/8538.html, 2021.

[16] Y. Evtushenko and M. Posypkin, *A deterministic algorithm for global multi-objective optimization*, Optimization Methods and Software, 29 (2013), pp. 1005–1019.

[17] Gurobi Optimization LLC, *Gurobi*. https://www.gurobi.com/. Accessed 2022-12-14.

[18] A. Göpfert, H. Riahi, C. Tammer, and C. Zalinescu, *Variational Methods in Partially Ordered Spaces*, Springer, 2003.

[19] P. T. Hoai, H. A. L. Thi, and N. C. Nam, *Half-open polyblock for the representation of the search region in multiobjective optimization problems: its application and computational aspects*, 4OR, 19 (2021), pp. 41–70.

[20] G. Kirlik and S. Sayin, *Bilevel programming for generating discrete representations in multiobjective optimization*, Mathematical Programming, 169 (2017), pp. 585–604.

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

[22] K. Klamroth, J. Tind, and M. M. Wiecek, *Unbiased approximation in multicriteria optimization*, Mathematical Methods of Operations Research (ZOR), 56 (2003), pp. 413–437.

[23] MATLAB, *Matlab bench documentation*. https://www.mathworks.com/help/matlab/ref/bench.html. Accessed 2022-12-14.

[24] S. Nobakhtian and N. Shafiei, *A benson type algorithm for nonconvex multiobjective programming problems*, TOP, 25 (2017), pp. 271–287.

[25] L. Paquete, B. Schulze, M. Stiglmayr, and A. C. Lourenço, *Computing representations using hypervolume scalarizations*, Computers & Operations Research, 137 (2022). 105349.

[26] A. Pascoletti and P. Serafini, *Scalarizing vector optimization problems*, Journal of Optimization Theory and Applications, 42 (1984), pp. 499–524.

[27] A. Przybylski, K. Klamroth, and R. Lacour, *A simple and efficient dichotomic search algorithm for multi-objective mixed integer linear programs*. https://arxiv.org/abs/1911.08937, 2019.

[28] S. Ruzika and M. M. Wiecek, *Approximation methods in multiobjective programming*, Journal of Optimization Theory and Applications, 126 (2005), pp. 473–501.

[29] L. Shao and M. Ehrgott, *An objective space cut and bound algorithm for convex multiplicative programmes*, Journal of Global Optimization, 58 (2013), pp. 711–728.

[30] ——, *Discrete representation of non-dominated sets in multi-objective linear programming*, European Journal of Operational Research, 255 (2016), pp. 687–698.

[31] K. Yang, M. Emmerich, A. Deutz, and T. Bäck, *Efficient computation of expected hypervolume improvement using box decomposition algorithms*, Journal of Global Optimization, 75 (2019), pp. 3–34.

[32] Ö. Özpeynirci and M. Köksalan, *An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs*, Management Science, 56 (2010), pp. 2302–2315.