

A Solver for Multiobjective Mixed-Integer Convex and Nonconvex Optimization

Gabriele Eichfelder · Oliver Stein · Leo Warnow

September 26, 2022

Abstract This paper proposes a general framework for solving multiobjective nonconvex optimization problems, i.e., optimization problems in which multiple objective functions have to be optimized simultaneously. Thereby, the nonconvexity might come from the objective or constraint functions, or from integrality conditions for some of the variables. In particular, multiobjective mixed-integer convex and nonconvex optimization problems are covered and form the focus of our studies.

The presented algorithm is based on a branch-and-bound method in the pre-image space, a technique which was already successfully applied for continuous nonconvex multiobjective optimization. We discuss how this basic method can be combined with appropriate branching rules and lower bounding procedures such that it is also implementable and convergent for multiobjective mixed-integer optimization problems.

For improving the performance of this new branch-and-bound method we enhance it with two types of cuts in the image space which are based on ideas from multiobjective mixed-integer convex optimization. Those combine continuous convex relaxations with adaptive cuts for the convex hull of the mixed-integer image set, derived from supporting hyperplanes to the relaxed sets.

Based on the above ingredients, the paper provides a new multiobjective mixed-integer solver for convex problems with a stopping criterion purely in

Gabriele Eichfelder

Institute of Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, E-mail: gabriele.eichfelder@tu-ilmenau.de, Orcid 0000-0002-1938-6316

Oliver Stein

Institute for Operations Research, Karlsruhe Institute of Technology (KIT), D-76128 Karlsruhe, Germany, E-mail: stein@kit.edu, Orcid 0000-0001-9514-6317, corresponding author

Leo Warnow

Institute of Mathematics, Technische Universität Ilmenau, Po 10 05 65, D-98684 Ilmenau, Germany, E-mail: leo.warnow@tu-ilmenau.de, Orcid 0000-0002-2177-8466

the image space. What is more, for the first time a solver for multiobjective mixed-integer nonconvex optimization is presented. We provide the results of numerical tests for the new algorithm. Where possible, we compare with existing procedures.

Keywords Branch-and-bound · Branch-and-cut · Multiobjective optimization · Mixed-integer optimization · Nonconvex optimization · Box enclosure

Mathematics Subject Classification (2000) 90C29 · 90C11 · 90C26

1 Introduction

We consider multiobjective mixed-integer optimization problems of the form

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0, \quad x \in X, \quad x_i \in \mathbb{Z}, \quad i \in I \quad (\text{MOMIP})$$

with a continuous vector-valued objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, a continuous vector-valued inequality constraint function $g : \mathbb{R}^n \rightarrow \mathbb{R}^k$, an n -dimensional box $X = [\underline{x}, \bar{x}]$ with $\underline{x}, \bar{x} \in \mathbb{R}^n$, $\underline{x} \leq \bar{x}$, and $I \subseteq \{1, \dots, n\}$ an index set of integer variables. The entries $\underline{x}_i, \bar{x}_i$, $i \in I$, are chosen from \mathbb{Z} without loss of generality. The inequality \leq as well as $<$ between vectors is always understood componentwise. We denote the set of feasible points by

$$M = M(X) = \{x \in X \mid x_i \in \mathbb{Z}, \quad i \in I, \quad g(x) \leq 0\}. \quad (1)$$

In case of $I \neq \emptyset$ the problem *MOMIP* is a nonconvex optimization problem, which makes it difficult to develop appropriate solution methods for this class of problems. Still, in case all appearing functions are convex, we call *MOMIP* a multiobjective mixed-integer convex optimization problem, otherwise a multiobjective mixed-integer nonconvex optimization problem.

The values of the vector-valued objective function will be compared componentwise which leads to the standard concept of efficiency (in the pre-image space) and nondominance (in the image space), as widely used in multiobjective optimization, see, for instance, [13]. As a consequence, there can be an infinite number of optimal values, i.e., of nondominated points, in the image space. This adds significant difficulties compared to the single-objective case ($m = 1$): in convex mixed-integer single-objective optimization with, for instance, $I = \{1\}$ and $x_1 \in \{0, 1\}$, just two single-objective continuous convex optimization problems have to be solved (one for each possible assignment of x_1) and the optimal values have to be compared. In the multiobjective setting this already leads to two convex multiobjective optimization problems for which no method is available that computes their full set of nondominated points. Moreover, in the end, sets of nondominated points have to be compared.

In case of $I = \emptyset$ the problem *MOMIP* collapses to a continuous multiobjective optimization problem *MOP*. If one or several of the functions f_i , $i \in \{1, \dots, m\}$ and g_j , $j \in \{1, \dots, k\}$ are nonconvex, the problem *MOP* is

denoted a nonconvex multiobjective optimization problem. Deterministic solution methods for nonconvex continuous multiobjective optimization problems have already been proposed, for instance, in [20,35] and, more recently, in [15].

We remark that any mixed-integer problem *MOMIP* could be artificially reformulated as a continuous problem *MOP* by introducing inequalities whose solution set is the integer grid. From this point of view, for instance the algorithm from [15] could also be directly applied to the mixed-integer setting. However, such continuous reformulations of mixed-integer problems are known to impede the algorithmic treatment in general. For example, the algorithm from [15] and its convergence proof would have to rely on an artificial and algorithmically expensive lower bounding technique using this approach.

Also, we will not try to determine all feasible integer assignments for the problem *MOMIP* and then to solve all related problems of type *MOP*. There is no method available which allows to calculate such a set of feasible integer assignments. Moreover, this would lead to too many problems of the type *MOP* which all have to be solved although their individual nondominated sets might not at all or only partially contribute to the set of nondominated points of *MOMIP*. We aim for a more direct iterative method which does not explore the individual integer assignments.

The aim of this paper is to propose an algorithm for the case $I \neq \emptyset$, for convex or nonconvex functions, which guarantees to stop after a finite number of iterations with an approximation of the nondominated set of certain quality. More precisely, we use the concept of an enclosure to cover the nondominated set and use its width as quality criterion. This concept was proposed in [15] as a generalization of the well-known gap concept from single-objective global optimization, where algorithms stop as soon as an interval of a certain length is determined which contains the optimal value. Our algorithm not only computes an enclosure of the nondominated set, but also, as a by-product, delivers a finite list of attainable points (i.e., images of feasible points), which contributes to the definition of the enclosure and which also serves as an approximation of the nondominated set of the problem *MOMIP*. However, the quality of the distribution of these points is not steered directly, as it is done for instance in [24] by using bilevel formulations. Instead, we solely aim to improve the quality of the enclosure.

As already mentioned, in this paper we do not artificially reformulate *MOMIP* as a continuous problem. Instead, we take the general branch-and-bound approach from [15] only as a basis and handle integrality conditions in a direct manner. We will then be able to use standard relaxations in the lower bounding techniques, and an adequate branching rule is simple to state and to execute. However, unfortunately, these constructions interfere with a crucial assumption for the convergence proof in the continuous setting from [15]. As a main contribution of the present article, by an appropriate modification of this assumption we will be able to show finite termination of the branch-and-bound method also for mixed-integer problems.

Moreover, we demonstrate that the proposed branch-and-bound method is not only conceptual but actually implementable and of practical use. Fur-

thermore, if continuous convex relaxations are employed, one may introduce polyhedral lower bounding sets which are adaptively refined by two types of cuts. Such cuts have been introduced in [11] for multiobjective mixed-integer convex optimization, but they can still be used for nonconvex problems in case convex relaxations are applied first. Since we apply these cuts in node problems of the branch-and-bound tree, this results in a branch-and-cut method. In contrast to cuts in the pre-image space, as they are commonly used in single-objective branch-and-cut methods, these are image-space cuts. We evaluate the benefits of these polyhedral lower bounds numerically.

We point out that our method is neither a pure pre-image nor a pure image space method. While it performs branching steps in the pre-image space, the node selection and termination criteria as well as the mentioned cuts are image space concepts.

For *convex* multiobjective mixed-integer optimization problems several solution approaches have already been presented in the literature. For example, a branch-and-cut algorithm is proposed in [11]. However, rather than using any derived lower bounds for determining a final enclosure, it just uses rough bounds from Lipschitz constants. Instead, the width of sub-boxes in the pre-image space is used as a stopping criterion which may lead to many boxes and thus limits the size of problems which can be treated. This is overcome with the method in [17] which is an algorithm working purely in the image space. It uses as an ingredient linear relaxations of the convex functions and is thus not, at least not immediately, usable for extensions to the nonconvex setting. We compare the performance of our algorithm with the ones of both these algorithms for the special case of convex mixed-integer optimization problems.

Specifically for the biobjective case (i.e., $m = 2$), an approach to solve mixed-integer convex optimization problems has been presented in [8]. This approach requires information about those feasible integer assignments which are at least ‘promising’ to contribute to the final efficient set. Also for $m = 2$, in [12] an approach was presented which approximates the nondominated set by line segments steered by optimizing a quality measure known as hyper-volume. The algorithm requires knowledge about the so-called nadir point which is not easily calculable for $m \geq 3$. What is more, for the concept of line segments a suitable higher-dimensional analogue is not obvious.

There are also specifically designed algorithms for *linear* multiobjective mixed-integer optimization. As non-linear functions are the focus of our contribution, our algorithm is not meant to explicitly exploit polyhedral structures. For completeness, we mention here some exemplary solvers, as for $m = 2$ [37] and for an arbitrary number of linear objectives [39].

For the first time, this paper proposes a solver for general *nonconvex* multiobjective mixed-integer optimization problems with a performance guarantee. Of course, it would be possible to apply scalarization techniques as proposed in [5] and then to solve the resulting parameter-dependent single-objective mixed-integer subproblems with available single-objective methods. However, this approach is limited to problem classes for which corresponding subsolvers are available as, for instance, quadratic mixed-integer problems. Moreover, it

is an open question how to choose the set of parameters for three or more objective functions to guarantee an enclosure (or a similar approximation) with a certain quality.

Also in the single-objective setting, solvers for mixed-integer nonconvex optimization problems are still an active area of research. For an overview see [7]. Possible approaches are relaxations which may be based on factorable functions or convexification using bounds on the eigenvalues and combining those with branching in the pre-image space. Within this paper, we generalize some of these basic ideas to the multiobjective setting. We would like to mention that for just one objective function also fundamentally different approaches exist as for instance copositive reformulations, see [6]. However, for those it is known that they cannot be transferred to the multiobjective setting [14].

The paper is structured as follows. In Section 2 we recall the basics from multiobjective optimization and give the basic ingredients for a general multiobjective branch-and-bound framework derived from lower and upper bounding sets. In this section we also present the overall algorithm as well as the proposed branching rule. Section 3 provides the finiteness results for our algorithm based on a collection of assumptions, which we keep in a general form to allow to use the algorithm with various lower bounding and box partition procedures. These assumptions are further discussed in Section 4 for specific classes of problems as convex or quadratic mixed-integer problems. Section 5 discusses possibilities to speed up our algorithm by adding image-space cuts, and Section 6 concludes with our numerical studies.

2 Ingredients for the Algorithm

In the following we introduce all needed concepts from multiobjective optimization such as optimality notions and the concept of an enclosure, as well as all ingredients for our proposed algorithm, including suitable concepts for lower and upper bounds.

2.1 Optimality Notions for Multiobjective Problems

Throughout this paper we use multiobjective optimality notions based on the natural ordering cone \mathbb{R}_+^m . The following definition works for any set Y in the image space \mathbb{R}^m of f , but will subsequently be applied to the set $Y = f(M)$ of attainable points of *MOMIP*. We remark that the branch-and-bound algorithm proposed in this paper relies on constructions in the image space \mathbb{R}^m . For details and motivation of the following notions we refer to [13,17,28,30].

Definition 2.1 For some set $Y \subseteq \mathbb{R}^m$ let $\bar{y} \in Y$.

- a) The point \bar{y} is called a nondominated point of Y if there exists no $y \in Y$ with $y \leq \bar{y}$ and $y \neq \bar{y}$.

- b) For $\varepsilon > 0$, the point \bar{y} is called an ε -nondominated point of Y if there exists no $y \in Y$ with $y \leq \bar{y} - \varepsilon e$ and $y \neq \bar{y} - \varepsilon e$, where e denotes the all-ones vector.
- c) The sets Y_N and Y_N^ε of all nondominated and ε -nondominated points are called nondominated set and ε -nondominated set of Y , respectively.

The following definition collects the according notions in the pre-image space \mathbb{R}^n of *MOMIP*.

Definition 2.2 Let $\bar{x} \in M$ be a feasible point of *MOMIP* and let $Y = f(M)$.

- a) The point \bar{x} is called efficient for *MOMIP* if $f(\bar{x}) \in Y_N$ holds.
- b) For $\varepsilon > 0$, the point \bar{x} is called ε -efficient for *MOMIP* if $f(\bar{x}) \in Y_N^\varepsilon$ holds.
- c) The sets M_E and M_E^ε of all efficient and ε -efficient points are called efficient set and ε -efficient set of *MOMIP*, respectively.

Under our assumptions the set M_E is nonempty whenever M is nonempty [13]. Due to the inclusions $M_E \subseteq M_E^\varepsilon$ as well as $Y_N = f(M_E) \subseteq Y_N^\varepsilon = f(M_E^\varepsilon)$ for any $\varepsilon > 0$, all of these sets are then nonempty.

In multiobjective optimization one aims in general to find the set of all nondominated points. Whenever this set is infinite, one has to be satisfied with an approximation of it which can, for instance, consist of a representation with a finite number of nondominated points as it was done in [24]. We aim on finding what is known as enclosure and which we define in Subsection 2.2.1.

2.2 A General Branch-and-Bound Framework

We aim on providing a natural generalization of the branch-and-bound ideas from the single-objective case. For continuous nonconvex multiobjective optimization problems a branch-and-bound framework was provided in [15]. This is in contrast to other multiobjective branch-and-bound methods as the one in [35] where lower bounds are just used for discarding and not as stopping criterion. In fact, with regard to its structure the subsequently stated Algorithm 1 coincides with [15, Algorithm 1], where the input *MOP* is replaced by *MOMIP*, and where the choice of the branching rule is to be specified separately. Our contribution will be to examine necessary assumptions for the lower bounding procedures and for the branching rule such that we are able to prove convergence even in the presence of integer variables, as the assumptions from [15] can no longer be satisfied in this case. Of course, we will also propose procedures which satisfy these assumptions. In the following, we explain the main ingredients that are necessary to understand the structure of Algorithm 1, which we state in Subsection 2.2.6.

2.2.1 The Enclosure of the Nondominated Set

In single-objective branch-and-bound the minimal value v of a function $f: M \rightarrow \mathbb{R}^1$ is sandwiched between successively improved lower bounds lb and upper

bounds ub . For a given tolerance $\varepsilon > 0$ the method terminates when the gap $ub - lb$ between lb and ub drops below ε . The upper bounds ub are generated as values $f(x)$ of successively updated feasible points $x \in M$. Upon termination the final incumbent $x \in M$ is ε -minimal in the sense that $v \leq f(x) \leq v + \varepsilon$ holds.

In the multiobjective setting the optimal value v corresponds to the non-dominated set Y_N of *MOMIP*. One may then generalize the values $lb, ub \in \mathbb{R}^1$ to lower and upper bounding sets $LB, UB \subseteq \mathbb{R}^m$ with

$$Y_N \subseteq (LB + \mathbb{R}_+^m) \cap (UB - \mathbb{R}_+^m), \quad (2)$$

where here and in the following sums of sets are meant in the Minkowski sense. This sandwiching approach for Y_N is in line with the ones reviewed in [38,41]. The enclosing interval $[lb, ub]$ for v from the single-objective case thus generalizes to the enclosure

$$E(LB, UB) := (LB + \mathbb{R}_+^m) \cap (UB - \mathbb{R}_+^m)$$

for Y_N . For illustrations of this concept see Figures 3 and 4.

2.2.2 The Gap Between Lower and Upper Bounding Sets

Having defined a suitable extension of the concept of an enclosing interval, it remains to define a suitable extension of the gap $ub - lb$. In [15] it is suggested to generalize this gap to the geometrical width

$$w(LB, UB) = \sup\{\|(y + te) - y\|_2 / \sqrt{m} \mid t \geq 0, y, y + te \in E(LB, UB)\}$$

of $E(LB, UB)$ in the direction of the all-ones vector e . Before we state a more explicit formula for this width, let us mention that without any further assumption, [15, Lem. 3.1] guarantees that for any $\varepsilon > 0$ and $w(LB, UB) < \varepsilon$ all attainable points of *MOMIP* in the enclosure $E(LB, UB)$ are ε -nondominated. In particular any algorithmically generated $x \in M$ with $f(x) \in E(LB, UB)$ is thus ε -efficient for *MOMIP*, which is in line with the ε -minimality of the final incumbent from the single-objective case. From this point of view, the above proposal for measuring the multiobjective gap is a natural concept.

It is not hard to see that for any lower and upper bounding sets LB, UB the enclosure $E(LB, UB)$ can be written as the union of the (possibly infinitely many) nonempty boxes which can be constructed with lower and upper bound vectors from LB and UB , respectively,

$$E(LB, UB) = \bigcup_{\substack{(\ell b, ub) \in LB \times UB \\ \ell b \leq ub}} [\ell b, ub]. \quad (3)$$

With

$$s(\ell b, ub) := \min_{j=1, \dots, m} (ub_j - \ell b_j) \quad (4)$$

denoting the length of a shortest edge of a box $[\ell b, ub]$, in [15, Lem. 3.2] it is shown that the geometrical width $w(LB, UB)$ of $E(LB, UB)$ in the direction

of the all-ones vector coincides with the supremum of $s(\ell b, ub)$ over all boxes $[\ell b, ub]$ from (3). We record this in the following definition.

Definition 2.3 For any sets $LB, UB \subseteq \mathbb{R}^m$ with (2) we call the supremum $w(LB, UB)$ of the problem

$$\max_{\ell b, ub} s(\ell b, ub) \quad \text{s.t.} \quad (\ell b, ub) \in LB \times UB, \ell b \leq ub$$

gap between LB and UB .

Since the supremum in Definition 2.3 is attained for nonempty and compact sets LB and UB ([15, Lem. 3.3]), we arrive at the following main result about the multiobjective gap.

Theorem 2.1 [15, Th. 3.4] For nonempty and compact sets $LB, UB \subseteq \mathbb{R}^m$ with (2) and some $\varepsilon > 0$ let the gap between LB and UB satisfy

$$\max \{s(\ell b, ub) \mid (\ell b, ub) \in LB \times UB, \ell b \leq ub\} < \varepsilon. \quad (5)$$

Then $E(LB, UB) \cap f(M) \subseteq Y_N^\varepsilon$ holds.

Our Algorithm 1 generates finite sets LB and UB , so that (3) yields a description of $E(LB, UB)$ by finitely many boxes, which we shall refer to as a *box enclosure* of Y_N . In this case the maximum in (5) is taken over finitely many boxes.

2.2.3 Provisional Nondominated Sets and Upper Bounding Sets

As in the single-objective case, upper bounding sets UB for Y_N are constructed from information at attainable points $f(x_{ub})$ with $x_{ub} \in M$ of *MOMIP*. However, in the multiobjective setting there is not a single incumbent, but different attainable points may provide good approximations for different nondominated points. Algorithm 1 thus keeps a finite list \mathcal{F} of attainable points generated so far.

Each time that some new point $x_{ub} \in M$ is generated in the course of the algorithm, its image $f(x_{ub})$ is only inserted into \mathcal{F} if $f(x_{ub})$ is not dominated by any element from \mathcal{F} . Moreover, all elements of \mathcal{F} which are dominated by $f(x_{ub})$ are deleted from \mathcal{F} . In the following we will refer to this procedure as updating \mathcal{F} with respect to $f(x_{ub})$. The source of the underlying feasible points x_{ub} will be elements of subboxes X' of X which are chosen for the discarding tests described below. If such an $x_{ub} \in X'$ is feasible for *MOMIP* the list \mathcal{F} will be updated with respect to $f(x_{ub})$.

As a consequence of this construction, \mathcal{F} forms a finite and stable subset of the set of attainable points of *MOMIP*, where stability means that no element of \mathcal{F} dominates any other element of \mathcal{F} . Since Y_N possesses the same property, \mathcal{F} is called *provisional nondominated set* [20]. In contrast to the single-objective case, setting $UB = \mathcal{F}$ does not provide a valid upper bounding set. To see this, just take, for example, for \mathcal{F} a single nondominated point

$\bar{y}^1 \in Y_N$, and further assume that the nondominated set contains also another nondominated point $\bar{y}^2 \in Y_N$. Then it does not hold $\bar{y}^2 \in \mathcal{F} - \mathbb{R}_+^m$ but just $\bar{y}^2 \notin \mathcal{F} + \mathbb{R}_+^m$. Still the information from \mathcal{F} may be used to construct an appropriate upper bounding set, as explained next (see also [35]).

The construction assumes the existence of a sufficiently large box $Z = [\underline{z}, \bar{z}]$ with $f(M) \subseteq \text{int}(Z)$, where int denotes the topological interior. In the present setting of *MOMIP* the set $f(M)$ is contained in the compact set $f(X)$, so that the existence of such a box Z is no restriction, and it may be computed by, e.g., interval arithmetic [23,31,40], see also [15].

Given a provisional nondominated set \mathcal{F} , the elements of the nondominated set Y_N cannot be dominated by any $q \in \mathcal{F}$. Hence $Y_N \setminus \mathcal{F}$ is a subset of the *search region*

$$S(\mathcal{F}) = \{z \in \text{int}(Z) \mid \forall q \in \mathcal{F} : q \not\leq z\} = \text{int}(Z) \setminus \left(\bigcup_{q \in \mathcal{F}} \{q\} + \mathbb{R}_+^m \right),$$

that is, the set of all points in $\text{int}(Z) \setminus \mathcal{F}$ which are not dominated by any point from \mathcal{F} . This implies $Y_N \subseteq \mathcal{F} \cup S(\mathcal{F})$. An algorithmically useful description of $S(\mathcal{F})$ is provided by the concept of *local upper bounds*.

Definition 2.4 ([26]) Let \mathcal{F} be a finite and stable subset of $f(M)$. A set $\text{lub}(\mathcal{F}) \subseteq Z$ is called local upper bound set with respect to \mathcal{F} if

- (i) $\forall z \in S(\mathcal{F}) : \exists p \in \text{lub}(\mathcal{F}) : z < p$,
- (ii) $\forall z \in (\text{int}(Z)) \setminus S(\mathcal{F}) : \forall p \in \text{lub}(\mathcal{F}) : z \not\leq p$,
- (iii) $\forall p_1, p_2 \in \text{lub}(\mathcal{F}) : p_1 \not\leq p_2$ or $p_1 = p_2$.

The set $\text{lub}(\mathcal{F})$ is uniquely determined, nonempty and finite, with $S(\mathcal{F}) = \bigcup_{p \in \text{lub}(\mathcal{F})} \{z \in \text{int}(Z) \mid z < p\}$ (cf. [15,26] for motivation and graphical illustration). Since by [35, Lem. 3.4] also $\mathcal{F} \subseteq \text{lub}(\mathcal{F}) - \mathbb{R}_+^m$ holds, one obtains $Y_N \subseteq \mathcal{F} \cup S(\mathcal{F}) \subseteq \text{lub}(\mathcal{F}) - \mathbb{R}_+^m$ and may define the upper bounding set $UB := \text{lub}(\mathcal{F})$. In the course of Algorithm 1, methods proposed in [10,26] may be employed to efficiently calculate and update the sets $\text{lub}(\mathcal{F})$ with respect to the appearing provisional nondominated sets \mathcal{F} .

2.2.4 Discarding by Relaxed Partial Upper Image Sets and Local Upper Bounds

In the single-objective case the overall lower bound lb for the minimal value v is defined as $lb = \min_{X' \in \mathcal{L}} lb'$ with partial lower bounds lb' for f on subsets $M(X')$, $X' \in \mathcal{L}$, of $M(X)$. Here X' denotes a subbox of X resulting from successive branching steps (cf. Sec. 2.2.7). The algorithm keeps a list \mathcal{L} of boxes X' for which $M(X')$ may contain globally minimal points. In fact, if lb' is a lower bound for f on the partial set $M(X')$ satisfying $lb' > ub = f(x_{ub})$ for the incumbent $x_{ub} \in M(X)$ and the resulting overall upper bound ub , then X' may be discarded from \mathcal{L} since $M(X')$ cannot contain any globally minimal point of f on $M(X)$. Partial lower bounds lb' are used in place of the

actual infimum of f on $M(X')$ since the computation of the latter is rarely algorithmically tractable.

In the multiobjective setting, for a subbox X' of X we call any compact set $LB' \subseteq \mathbb{R}^m$ with $f(M(X')) + \mathbb{R}_+^m \subseteq LB' + \mathbb{R}_+^m$ *partial lower bounding set* for $f(M(X'))$, that is, the upper set $LB' + \mathbb{R}_+^m$ of LB' provides a relaxation for the upper image set of f on $M(X')$. In the continuous nonconvex case from [15] interval arithmetic [23,31,40], the α BB method [1,2] as well as reformulation-linearization techniques (RLT) [3,42–44] are discussed as examples for convex relaxation techniques. In the present setting of the nonconvex and mixed-integer problem *MOMIP*, such techniques may either be applied directly, leading to mixed-integer convex or mixed-integer linear relaxations, or in combination with continuous relaxation, which yields continuous convex or continuous linear relaxations. Our framework also covers other relaxation techniques, as long as they lead to algorithmically tractable relaxed problems and the assumptions for convergence from Section 3 hold. For a given choice T of a relaxation technique we denote the resulting partial lower bounding sets by LB'_T .

We point out that the above-mentioned relaxation techniques T usually work in the pre-image space \mathbb{R}^n for the construction of relaxations $M_T(X') \subseteq X'$ of $M(X')$, and in the combined pre-image and image space $\mathbb{R}^n \times \mathbb{R}^m$ for the construction of relaxations f_T of f on X' . In this way they provide a relaxation $\text{epi}(f_T, M_T(X'))$ of the epigraph $\text{epi}(f, M(X')) := \{(x, y) \in M(X') \times \mathbb{R}^m \mid f(x) \leq y\}$ of f on $M(X')$. The image-set object $LB'_T + \mathbb{R}_+^m$ then formally results by parallel projection of $\text{epi}(f_T, M_T(X'))$ to \mathbb{R}^m .

For a subbox $X' \subseteq X$ with $M(X') \neq \emptyset$, the point $a' \in \mathbb{R}^m$ with components

$$a'_j = \min_{x \in M(X')} f_j(x), \quad j = 1, \dots, m, \quad (6)$$

is called *ideal point* of the partial image set $f(M(X'))$. Lower estimates for ideal points will play a crucial role in the sequel. In fact, for each relaxation technique T and partial lower bounding set LB'_T of $f(M(X'))$ the infima of the problems

$$\min_y y_j \quad \text{s.t.} \quad y \in LB'_T + \mathbb{R}_+^m \quad (7)$$

with $j \in \{1, \dots, m\}$ form a vector $\tilde{a}'_T \leq a'$, that is, the choice of the relaxation technique T induces a specific lower estimate for the ideal point of $f(M(X'))$ via the set LB'_T . With these constructions the discarding test from the single-objective case can be generalized as follows.

Theorem 2.2 [15, Cor. 5.11] *Let \mathcal{F} be a finite and stable subset of $f(M)$, let X' be a subbox of X , let LB'_T be some partial lower bounding set for $f(M(X'))$, and let \tilde{a}'_T be the induced lower estimate of the ideal point of f on $M(X')$. If*

$$p \notin LB'_T + \mathbb{R}_+^m \quad (8)$$

*holds for all $p \in \text{lub}(\mathcal{F}) \cap (\tilde{a}'_T + \mathbb{R}_+^m)$, then $f(M(X'))$ does not contain any nondominated point of *MOMIP* and X' can be discarded from \mathcal{L} .*

Figure 1 illustrates a case in which X' can be discarded using Theorem 2.2 and a continuous convex relaxation. More precisely, the box can be discarded since no local upper bound $p \in \text{lub}(\mathcal{F})$ (indicated by crosses) is contained in the upper image of LB'_T . As stated in Theorem 2.2, it is actually sufficient that the latter only holds for $p \in \text{lub}(\mathcal{F}) \cap (\tilde{a}'_T + \mathbb{R}_+^m)$.

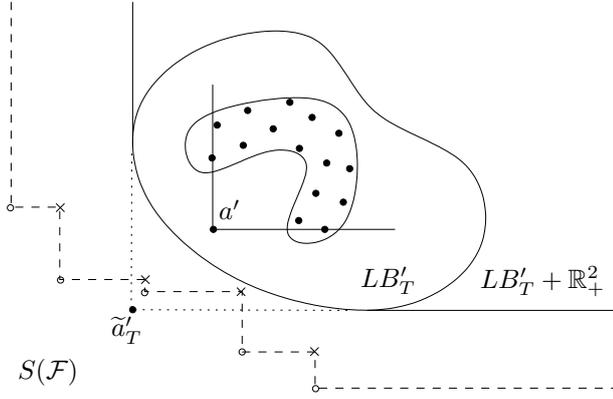


Fig. 1: Discarding test with $f(M(X'))$ (filled dots), \mathcal{F} (circles) and $\text{lub}(\mathcal{F})$ (crosses)

Condition (8) is equivalent to the positivity of the optimal value $\varphi_{LB'_T}(p)$ of the problem

$$\min_t t \quad \text{s.t.} \quad p + te \in LB'_T + \mathbb{R}_+^m \quad (9)$$

(cf. [15, Lem. 5.5] and [9,21]), exhibiting a similar constraint structure as (7). For the algorithmic treatment of (7) and (9), their image space formulation is complemented by pre-image space variables x , and using the epigraphical formulation they may be equivalently replaced by

$$\min_x (f_T)_j(x) \quad \text{s.t.} \quad x \in M_T(X') \quad (10)$$

and

$$\min_{x,t} t \quad \text{s.t.} \quad f_T(x) \leq p + te, \quad x \in M_T(X'), \quad (11)$$

respectively. By computing the ideal point estimate \tilde{a}'_T as the vector of optimal values of the problems in (10), $j \in \{1, \dots, m\}$, and by computing $\varphi_{LB'_T}(p)$ as the optimal value of (11), one obtains an algorithmically tractable discarding test based on Theorem 2.2.

2.2.5 Overall Lower Bounding Sets and Box Enclosures

While for a subbox X' of X and a relaxation technique T the discarding test from Theorem 2.2 combines the coarse partial lower bounding set \tilde{a}'_T with the

tighter lower bounding set LB'_T , [15] proposes to construct an overall lower bounding set for Y_N only from the ideal point estimates \tilde{a}'_T . In fact, let $A_{T,N}$ denote the set of nondominated points of $\{\tilde{a}'_T \mid X' \in \mathcal{L}\}$. Then $LB_T := A_{T,N}$ is an overall lower bounding set for Y_N which even satisfies $Y_N \cup \mathcal{F} \subseteq LB_T + \mathbb{R}_+^m$ [15, Lem. 6.5]. Altogether we obtain the box enclosure

$$E(A_{T,N}, \text{lub}(\mathcal{F})) = \bigcup_{\substack{(a,p) \in A_{T,N} \times \text{lub}(\mathcal{F}) \\ a \leq p}} [a, p]$$

of $Y_N \cup \mathcal{F}$, and in view of Theorem 2.1 the following result holds.

Theorem 2.3 [15, Th. 7.1] *In some iteration of the branch-and-bound method let $A_{T,N}$ be the nondominated set of the current set of induced ideal point estimates $\{\tilde{a}'_T \mid X' \in \mathcal{L}\}$, let \mathcal{F} denote the current provisional nondominated set, and for some $\varepsilon > 0$ let the gap between $A_{T,N}$ and $\text{lub}(\mathcal{F})$ satisfy*

$$\max \{s(a, p) \mid (a, p) \in A_{T,N} \times \text{lub}(\mathcal{F}), a \leq p\} < \varepsilon. \quad (12)$$

Then all $q \in \mathcal{F}$ are ε -nondominated points of MOMIP, and $E(A_{T,N}, \text{lub}(\mathcal{F}))$ forms a box enclosure of Y_N with width $w(A_{T,N}, \text{lub}(\mathcal{F})) < \varepsilon$.

2.2.6 Pseudocode of the Main Algorithm

Based on the discussed ingredients we are now able to present the structure of our main procedure in Algorithm 1, named MOMIBB. It remains to clarify the branching rule in line 11 which we discuss in the subsequent Subsection 2.2.7 in detail.

2.2.7 Node Selection and Branching

If for $\varepsilon > 0$ the condition (12) is violated, then the gap between $A_{T,N}$ and $\text{lub}(\mathcal{F})$ is reduced by branching a box $X^* \in \mathcal{L}$. For this purpose, we chose a box $X^* \in \mathcal{L}$ for which there is a lower estimate \tilde{a}_T^* for the ideal point of the partial image set $f(M(X^*))$ with $\tilde{a}_T^* \in A_{T,N}$ and for which it holds for some $p^* \in \text{lub}(\mathcal{F}) \cap (\tilde{a}_T^* + \mathbb{R}_+^m)$ that the length of the shortest edge of the box $[\tilde{a}_T^*, p^*]$ equals the gap between $A_{T,N}$ and $\text{lub}(\mathcal{F})$, i.e., with

$$s(\tilde{a}_T^*, p^*) = \max \{s(a, p) \mid (a, p) \in A_{T,N} \times \text{lub}(\mathcal{F}), a \leq p\}.$$

The finite termination and performance of the resulting Algorithm 1 mainly rely on the choice of an adequate branching rule in line 11 and an appropriate lower bounding technique T for lines 14, 15 and 16. We propose a generalization of the branching rule from [15] (which was used already in [11]) in Algorithm 2 and will discuss lower bounding techniques in Section 3.

Before presenting this branching rule, we remark that improvements of Algorithm 1 regarding its memory usage are possible by introducing additional filtering steps. For example, after updating the set of local upper bounds in

Algorithm 1 Multiobjective Mixed-Integer Branch-and-Bound Method (MOMIBB)

Input: Problem *MOMIP* with $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^k$, $I \subseteq \{1, \dots, n\}$, $X = [\underline{x}, \bar{x}]$, $\underline{x}, \bar{x} \in \mathbb{R}^n$, $\underline{x} < \bar{x}$, $x_i, \bar{x}_i \in \mathbb{Z}$, $i \in I$, termination tolerance $\varepsilon > 0$, partial lower bounding technique T .

- 1: Compute a box $Z = [\underline{z}, \bar{z}]$ with $f(M(X)) \subseteq \text{int}(Z)$.
 - 2: Initialize list $\mathcal{L}^0 \leftarrow \{(X, \{\underline{z}\}, \underline{z})\}$.
 - 3: Initialize set of ideal point estimates $A_T^0 \leftarrow \{\underline{z}\}$.
 - 4: Initialize provisional nondominated set $\mathcal{F}^0 \leftarrow \emptyset$.
 - 5: Initialize set of local upper bounds $\text{lub}(\mathcal{F}^0) = \{\bar{z}\}$.
 - 6: Initialize iteration counter $k \leftarrow 0$.
 - 7: **repeat**
 - 8: $k \leftarrow k + 1$.
 - 9: Choose $(X^k, LB_T^k, \tilde{a}_T^k)$ from \mathcal{L}^{k-1} with $\tilde{a}_T^k \in A_T^{k-1}$
and $p^k \in \text{lub}(\mathcal{F}^{k-1}) \cap (\tilde{a}_T^k + \mathbb{R}_+^m)$ with
 $s(\tilde{a}_T^k, p^k) = \max\{s(a, p) \mid (a, p) \in A_T^{k-1} \times \text{lub}(\mathcal{F}^{k-1}), a \leq p\}$.
 - 10: $\mathcal{L}^k \leftarrow \mathcal{L}^{k-1} \setminus \{(X^k, LB_T^k, \tilde{a}_T^k)\}$.
 - 11: Divide X^k into $X^{k,1}$ and $X^{k,2}$ by a branching rule.
 - 12: $\mathcal{F}^k \leftarrow \mathcal{F}^{k-1}$, $\text{lub}(\mathcal{F}^k) \leftarrow \text{lub}(\mathcal{F}^{k-1})$.
 - 13: **for** $\ell = 1, 2$ **do**
 - 14: Determine a partial lower bounding set $LB_T^{k,\ell}$ for $f(M(X^{k,\ell}))$.
 - 15: Determine the ideal point estimate $\tilde{a}_T^{k,\ell}$ induced by $LB_T^{k,\ell}$.
 - 16: **if** there is some $p \in \text{lub}(\mathcal{F}^k) \cap (\tilde{a}_T^{k,\ell} + \mathbb{R}_+^m)$ with $\varphi_{LB_T^{k,\ell}}(p) \leq 0$ **then**
 - 17: $\mathcal{L}^k \leftarrow \mathcal{L}^k \cup \{(X^{k,\ell}, LB_T^{k,\ell}, \tilde{a}_T^{k,\ell})\}$.
 - 18: **if** some $x^{k,\ell} \in M(X^{k,\ell})$ is available **then**
 - 19: $\mathcal{F}^k \leftarrow$ update of \mathcal{F}^k with respect to $f(x^{k,\ell})$.
 - 20: $\text{lub}(\mathcal{F}^k) \leftarrow$ update of $\text{lub}(\mathcal{F}^k)$.
 - 21: **until** $\mathcal{L}^k = \emptyset$ or $\max\{s(a, p) \mid (a, p) \in A_T^k \times \text{lub}(\mathcal{F}^k), a \leq p\} < \varepsilon$.
 - Output:**
 - 22: **if** $\mathcal{L}^k \neq \emptyset$ **then**
 \mathcal{F}^k is a set of ε -nondominated points of *MOMIP*,
and $E(A_T^k, \text{lub}(\mathcal{F}^k))$ is a box enclosure of Y_N with $w(A_T^k, \text{lub}(\mathcal{F}^k)) < \varepsilon$.
 - 23: **else**
MOMIP is inconsistent.
-

line 20 for some of the subboxes $X' \in \mathcal{L}^k$ the condition (8) may hold for all $p \in \text{lub}(\mathcal{F}) \cap (\tilde{a}_T' + \mathbb{R}_+^m)$. Hence these boxes could be discarded, i.e., removed from the list \mathcal{L}^k . As a consequence of this filtering step, an advantage would be that the box selection in line 9 of the algorithm would become faster since there are less elements in the list \mathcal{L}^k to consider. However, in our numerical tests there was almost no noticeable difference in computation times for line 9 with and without the filtering. Hence, the additional computational effort for the filtering was usually not worth it. We remark that in Algorithm 1 the list entries of \mathcal{L}^k are triples consisting of subboxes X^k and corresponding partial lower bounding sets as well as ideal point estimators, which avoids possible re-computations of the latter two entries in the course of the algorithm. However, as above, by a slight abuse of notation we refer to the list entries as mere boxes if no confusion is possible.

Another filtering step can be included at the end of each iteration of the repeat loop. More precisely, in line with Theorem 2.3 it would be possible to reduce the set A_T^k to its stable set $A_{T,N}^k$, i.e., to remove all dominated points within the set. Again, this could make the selection of a new box in line 9 less time consuming. However, it turns out that also this improvement is almost neglectable for our numerical tests in Section 6 while the computational costs for adding the filtering step are indeed noticeable.

Algorithm 2 bisects a given box along a longest edge not only for the coordinate directions of continuous variables like in [15], but also if the coordinate direction corresponds to one of the integer variables x_i , $i \in I$. This is in contrast to a common branching rule in single-objective mixed-integer optimization, which branches on integer variables whose value in a computed optimal point of a continuously relaxed node problem is fractional. This is not possible in the present setting for several reasons. First, our approach does not only consider continuous relaxations but allows for alternative relaxations. Second, node problems are not ‘solved’ by Algorithm 1, since this would amount to the computation of the whole efficient and nondominated sets. Third, even if the whole efficient set was computed, a strategy would be needed to choose efficient points with fractional entries for branching. We point out that, while Algorithm 2 does not use such information from the node problem for branching, we will generate cuts from node information in Section 5.

The bisection along a longest edge in the case of an integer coordinate is refined in Algorithm 2 by a usual rounding rule. This involves line 9, which is well-defined for any input from Algorithm 1 since there no box with a longest edge length of zero is chosen for branching.

Algorithm 2 Branching rule

Input: Box $X = [\underline{x}, \bar{x}]$, $\underline{x}, \bar{x} \in \mathbb{R}^n$, $\underline{x} < \bar{x}$, index set $I \subseteq \{1, \dots, n\}$ with $\underline{x}_i, \bar{x}_i \in \mathbb{Z}$, $i \in I$.

- 1: Compute the index k of a longest edge $X_k = [\underline{x}_k, \bar{x}_k]$ of X .
- 2: $\text{mid}(X_k) \leftarrow \frac{1}{2}(\underline{x}_k + \bar{x}_k)$.
- 3: **if** $k \in I$ **then**
- 4: **if** $\bar{x}_k - \underline{x}_k$ is odd **then**
- 5: $X_k^1 \leftarrow [\underline{x}_k, \lfloor \text{mid}(X_k) \rfloor]$,
- 6: $X_k^2 \leftarrow [\lceil \text{mid}(X_k) \rceil, \bar{x}_k]$.
- 7: **else**
- 8: $X_k^1 \leftarrow [\underline{x}_k, \text{mid}(X_k)]$,
- 9: $X_k^2 \leftarrow [\text{mid}(X_k) + 1, \bar{x}_k]$.
- 10: **else**
- 11: $X_k^1 \leftarrow [\underline{x}_k, \text{mid}(X_k)]$,
- 12: $X_k^2 \leftarrow [\text{mid}(X_k), \bar{x}_k]$.
- 13: Construct X^1 and X^2 from X by replacing the edge X_k with the edges X_k^1 and X_k^2 , respectively.

Output: Subboxes X^1 and X^2 of X .

3 Finite Termination in the General Case

This section provides general assumptions under which Algorithm 1 terminates after finitely many steps.

3.1 Convergence Under General Assumptions

We start by recalling the convergence considerations from [15] and by adapting them to the mixed-integer setting.

The convergence proof from the continuous case relies on the convergence behavior of sequences of induced ideal point estimates (\tilde{a}_T^k) on exhaustive sequences of boxes (X^k) in X . Thereby, a sequence of boxes (X^k) in X is called exhaustive if $X^{k+1} \subseteq X^k \subseteq X$ holds for all $k \in \mathbb{N}$ and $\lim_k \text{diag}(X^k) = 0$, where $\text{diag}(X') = \|\bar{x}' - \underline{x}'\|_2$ denotes the diagonal length of a box $X' \subseteq X$. In a branch-and-bound framework, sequences of boxes (X^k) are generated by successive branching of the box X according to the given branching rule. One needs to ensure that any such sequence generated by the branching rule is exhaustive, see the forthcoming Assumption 3.1. Algorithm 2 satisfies it since it bisects boxes along a longest edge.

Assumption 3.1 *Any sequence of boxes (X^k) in X which is generated by infinitely many steps of the branching rule is exhaustive.*

For any exhaustive sequence of boxes (X^k) the set $\bigcap_{k \in \mathbb{N}} X^k$ is a singleton, say $\{\tilde{x}\}$. While \tilde{x} clearly lies in X , for the convergence of a branch-and-bound scheme the behaviour of the chosen lower bounding technique in the two cases $\tilde{x} \in M(X)$ and $\tilde{x} \notin M(X)$ is crucial. Since additional properties of \tilde{x} may be derived from the branching rule, the interplay between the lower bounding technique and the branching rule must be adequate, in a sense to be explained below.

Parts a and b of the following definition for single-objective problems are taken from [25] and adapted to the mixed-integer case.

Definition 3.1 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^1$, $g : \mathbb{R}^m \rightarrow \mathbb{R}^k$ and $M(X) = \{x \in X \mid x_i \in \mathbb{Z}, i \in I, g(x) \leq 0\}$.

- a) A function ℓ from the set of all subboxes X' of X to $\mathbb{R} \cup \{+\infty\}$ is called *M-dependent lower bounding procedure* if $\ell(X') \leq \inf_{x \in M(X')} f(x)$ holds for all subboxes $X' \subseteq X$.
- b) An *M-dependent lower bounding procedure* is called *convergent* if every exhaustive sequence of boxes (X^k) in X satisfies

$$\lim_k \ell(X^k) = \lim_k \inf_{x \in M(X^k)} f(x). \quad (13)$$

- c) Given a particular branch-and-bound algorithm A , an *M-dependent lower bounding procedure* is called *convergent for A* if every exhaustive sequence of boxes (X^k) generated by A satisfies (13).

Along the lines of [15] we connect this single-objective concept to the multiobjective setting as follows. In Algorithm 1 the entries $(\tilde{a}'_T)_j$ of the induced ideal point estimate for $f(M(X'))$ may be interpreted as the results $\ell_{T,j}(X')$ of a lower bounding procedure $\ell_{T,j}$ for f_j on $M(X')$. With $\ell_T(X')$ denoting the vector with entries $\ell_{T,j}(X')$ this yields $\tilde{a}'_T = \ell_T(X')$, so that the convergence behaviour of the ideal point estimates follows directly from the convergence behaviour of the lower bounding procedure.

While convergence of the lower bounding procedure in the sense of Definition 3.1b is assumed in the convergence proof from [15], it is not hard to see that this may be weakened to convergence for [15, Algorithm 1] in the sense of Definition 3.1c. The weakening of this assumption will be crucial subsequently and is stated as follows.

Assumption 3.2 *The entries of the induced ideal point estimates are computed by some M -dependent lower bounding procedure which is convergent for Algorithm 1.*

With regard to upper bounds, the following assumption makes sure that in line 18 of Algorithm 1 feasible points in sufficiently small boxes can be computed when they exist.

Assumption 3.3 *There exist some $\delta > 0$ and some procedure so that for all boxes $X' \subseteq X$ created by Algorithm 1 with $\text{diag}(X') < \delta$ and $M(X') \neq \emptyset$ a feasible point $x' \in M(X')$ can be computed.*

Theorem 3.1 [15, Th. 8.5] *Let Assumptions 3.1, 3.2 and 3.3 hold. Then for any $\varepsilon > 0$ Algorithm 1 terminates after a finite number of iterations.*

Observe that in [15] the above theorem is shown under the general Assumptions 3.2 and 3.3, but for a special branching rule, namely the purely continuous version of Algorithm 2. However, only its exhaustiveness property from Assumption 3.1 is used in the proof, so that Theorem 3.1 is a valid reformulation of [15, Th. 8.5].

We mention that [16, Cor. 5.2] proves convergence in the Hausdorff metric of the box enclosures of Y_N to the boundary of the upper image set $f(M(X)) + \mathbb{R}_+^m$ for ε tending to zero (relative to the box Z).

The applicability of Theorem 3.1 clearly depends on how restrictive Assumptions 3.2 and 3.3 are. We shall discuss this issue next.

3.2 Applicability of Assumption 3.2

While basically all lower bounding procedures that are commonly used in *continuous* global optimization are convergent in the sense of Definition 3.1b [15], convergence in this sense cannot be expected in the mixed-integer case, even for such a fundamental lower bounding technique T as continuous relaxation (for which we put $T := CR$). This is mainly due to the case $\tilde{x} \notin M(X)$ for an exhaustive sequence (X^k) with $\bigcap_{k \in \mathbb{N}} X^k = \{\tilde{x}\}$. Then we have $M(X^k) = \emptyset$ for

all sufficiently large k , so that with the usual convention $\inf_{x \in \emptyset} f(x) = +\infty$ the convergence of a lower bounding procedure requires $\lim_k \ell(X^k) = +\infty$.

On the other hand, the computation of the ideal point estimate \tilde{a}_{CR}^k amounts to the solution of the m single-objective continuous problems

$$\min f_j(x) \quad \text{s.t.} \quad g(x) \leq 0, \quad x \in X^k, \quad j = 1, \dots, m. \quad (14)$$

Assume that $\tilde{x} \notin M(X)$ holds due to $\tilde{x}_i \notin \mathbb{Z}$ for some $i \in I$, while there are points $x^k \in X^k$ with $g(x^k) \leq 0$ for all k . Then each x^k is feasible for the problems in (14), and in view of $\lim_k \text{diag}(X^k) = 0$ the sequence (x^k) converges to \tilde{x} . The continuity of f_j thus yields

$$\lim_k \tilde{a}_j^k = \lim_k \ell_{CR,j}(X^k) \leq \lim_k f_j(x^k) = f_j(\tilde{x}) < +\infty, \quad (15)$$

in contradiction to the required convergence property $\lim_k \ell_{CR,j}(X^k) = +\infty$.

For this reason we introduced convergence of a lower bounding procedure for an algorithm in Definition 3.1c. This allows us to infer additional properties of points \tilde{x} with $\bigcap_{k \in \mathbb{N}} X^k = \{\tilde{x}\}$ for an exhaustive sequence of boxes (X^k) generated by this algorithm. In particular, we may take effects of the chosen branching rule into account. Such a useful effect may be formulated as follows.

Assumption 3.4 *Let (X^k) be any exhaustive sequence in X which is generated by the branching rule. Then for all sufficiently large k , all integer variables x_i , $i \in I$, of vectors $x \in X^k$ are fixed to some constant integer value.*

Lemma 3.1 *The branching rule from Algorithm 2 satisfies Assumption 3.4.*

Proof Exhaustiveness of (X^k) implies that for all sufficiently large k the maximal edge length of $X^k = [\underline{x}^k, \bar{x}^k]$ lies strictly below one. The branching rule from Algorithm 2 then implies that each edge length $\bar{x}_i^k - \underline{x}_i^k$ with $i \in I$ is zero. This means that for all sufficiently large k all integer variables x_i , $i \in I$, of vectors $x \in X^k$ are fixed to the constant integer value $\bar{x}_i^k = \underline{x}_i^k$. \square

The following theorem shows in particular that the branching rule from Algorithm 2 prevents the described lack of convergence due to the presence of integer variables.

Theorem 3.2 *Let $M_{CR}(X) = \{x \in X \mid g(x) \leq 0\}$ denote the continuous relaxation of $M(X)$, let the branching rule satisfy Assumption 3.4, and for some component function f_j of f let ℓ be an M -dependent convergent lower bounding procedure for f_j on $M_{CR}(X)$. Then ℓ is an M -dependent lower bounding procedure for f_j on $M(X)$, and ℓ is convergent for Algorithm 1.*

Proof Due to $\inf_{x \in M_{CR}(X')} f_j(x) \leq \inf_{x \in M(X')} f_j(x)$ for any $X' \subseteq X$, ℓ is also an M -dependent lower bounding procedure for f_j on $M(X)$. For the proof of convergence let (X^k) be an exhaustive sequence of boxes generated by Algorithm 1. In view of Assumption 3.4, for all sufficiently large k the additional explicit statement of integrality conditions in $M(X^k)$ is superfluous, so that $M(X^k)$ coincides with its continuous relaxation $M_{CR}(X^k)$. As ℓ is convergent for f_j on $M_{CR}(X)$, this shows that ℓ is also convergent for f_j on $M(X)$. \square

Since, as mentioned before, basically all lower bounding procedures from continuous global optimization are convergent, their application to the continuous relaxation in combination with a branching rule satisfying Assumptions 3.1 and 3.4 leads to the validity of Assumption 3.2. This includes the combination of interval arithmetic, the α BB method or RLT (see [15] and the references therein) with Algorithm 2. For example, lower bounding by α BB convexification of the continuous relaxation leads to smooth convex problems (10) and (11).

3.3 Applicability of Assumption 3.3

In Assumption 3.3 the choice $\delta = 1$ as a strict upper bound for the diagonal length of a box X' implies that also all edge lengths of X' are strictly bounded from above by one. In combination with a branching rule satisfying Assumption 3.4 this fixes all integer variables to integer values by the same arguments as in the proof of Lemma 3.1. Hence the applicability question for this assumption reduces to the purely continuous case, which is discussed in [15, Sec. 8].

From a practical perspective, the generation of feasible points for larger boxes, for which integer variables are not yet fixed, may be supported by the arsenal of primal heuristics for mixed-integer optimization (cf. [4] for a survey). This also includes the granularity based primal heuristics for the linear [33], convex [32], and nonconvex [34] case.

4 Finite Termination for Specially Structured Problems

The convergence considerations from Section 3 simplify significantly if the problem *MOMIP* exhibits certain structurally favorable characteristics. One such characteristic is the absence of integer variables ($I = \emptyset$) which is discussed in [15]. Subsequently we briefly discuss explicit designs of Algorithm 1 for three other special cases, involving integer variables.

4.1 Convergence in the Convex Mixed-Integer Case

In this section we impose the following convexity assumption.

Assumption 4.1 *The component functions of f and g are convex and continuously differentiable on X .*

Following [11], a MOMIP satisfying Assumption 4.1 is called a MOMICP. For any MOMICP, choosing merely continuous relaxation as the partial lower bounding technique in Algorithm 1 already leads to efficiently computable ideal point estimates, since then the continuous problems in (14) are defined by smooth convex functions. This means that for any MOMICP we may use

the simple lower bounding procedure $\ell_{CR,j}(X') = \inf_{x \in M_{CR}(X')} f_j(x)$ in Theorem 3.2, and Assumption 3.2 is satisfied.

Regarding Assumption 3.3, from Section 3.3 we know that it suffices to be able to find a feasible point of the continuous relaxation $M_{CR}(X')$ for sufficiently small boxes X' . Since this amounts to the solution of smooth and convex feasibility problems, also Assumption 3.3 is satisfied. Theorem 3.1 thus yields the following result.

Corollary 4.1 (Finite termination for MOMICPs) *Let Assumption 4.1 hold and choose continuous relaxation as the lower bounding technique T (i.e., $T := CR$). Then for any $\varepsilon > 0$ Algorithm 1 with the branching rule from Algorithm 2 terminates after a finite number of iterations.*

Our numerical experience with MOMIBB (Algorithm 1) for MOMICPs is reported in Section 6.

4.2 Convergence in the Linear Mixed-Integer Case

While MILPs are hard to handle from a complexity perspective, state-of-the-art software can often solve them within reasonable time limits. Therefore, the following polyhedrality assumption is a favorable characteristic of MOMIPs.

Assumption 4.2 *The component functions of f and g are affine-linear on X .*

We refer to a MOMIP satisfying Assumption 4.2 as a MOMILP. For MOMILPs the ideal points of partial image sets do not have to be underestimated, but the computation of their entries by (6) amounts to the solution of m MILPs. Formally this corresponds to choosing the partial lower bounding set LB'_T to be identical to the partial image set $f(M(X'))$, which we indicate by $T := PIS$. Moreover, Assumption 3.3 only requires the solution of continuous linear feasibility problems for sufficiently small boxes. Theorem 3.1 hence implies the following corollary.

Corollary 4.2 (Finite termination for MOMILPs) *Let Assumption 4.2 hold and choose the lower bounding technique $T = PIS$. Then for any $\varepsilon > 0$ Algorithm 1 with the branching rule from Algorithm 2 terminates after a finite number of iterations.*

Since in this setting Algorithm 1 works without continuous convex relaxations, subsequently we shall refer to it as MOMIBB_{direct}.

4.3 Convergence in the Quadratic Mixed-Integer Case

Recent advances in software packages for mixed-integer optimization allow a certain generalization of the approach from Section 4.2. It uses the following weakening of Assumption 4.2.

Assumption 4.3 *The component functions of f and g are quadratic on X .*

We refer to a MOMIP satisfying Assumption 4.3 as a MOMIQP. State-of-the-art software packages like Gurobi [22] are able to handle quadratic mixed-integer optimization problems, where actually not even convexity of the quadratic functions is required. Proceeding without continuous relaxation, like in Section 4.2 we arrive at the following result.

Corollary 4.3 (Finite termination for MOMIQPs) *Let Assumption 3.3 and 4.3 hold and choose the lower bounding technique $T = PIS$. Then for any $\varepsilon > 0$ Algorithm 1 with the branching rule from Algorithm 2 terminates after a finite number of iterations.*

Unlike Corollary 4.2, Corollary 4.3 explicitly requires Assumption 3.3. This means that one must be able to solve continuous quadratic feasibility problems for sufficiently small boxes, which is guaranteed at least for convex quadratic component functions of g on X .

Our numerical experience with Algorithm 1 for MOMIQPs is reported in Section 6. Again, since in this setting Algorithm 1 works without continuous convex relaxations, it is of the type MOMIBB_{direct}. In contrast to Section 4.2, here quadratic rather than linear mixed-integer subproblems have to be solved.

5 Cuts in the Node Problems

This section discusses possibilities to speed up Algorithm 1 under the following assumption.

Assumption 5.1 *The discarding test problem (11) is continuous with the function f_T and the functions defining $M_T(X')$ being differentiable and convex.*

In the case when Assumption 5.1 holds, but the cuts presented in this section are not employed, we refer to Algorithm 1 as MOMIBB-c0. Assumption 5.1 holds, for example, if the lower bounding technique T combines continuous relaxation with convex relaxation like α BB, or if continuous relaxation is used under Assumption 4.1 (Cor. 4.1). Under Assumption 5.1 also the computation of the ideal point estimates from (10) amounts to smooth convex problems. However, the latter only requires the solution of m problems, while the number of smooth convex problems (11) in discarding tests may be vast, depending on the number and position of local upper bounds. More precisely, recall that the discarding test from Theorem 2.2 allows to discard the box X' if

$$p \notin LB'_T + \mathbb{R}_+^m \quad \forall p \in \text{lub}'(\mathcal{F}) := \text{lub}(\mathcal{F}) \cap (\tilde{a}'_T + \mathbb{R}_+^m) \quad (16)$$

holds. In line 16 of Algorithm 1 the condition (16) is checked by its equivalent reformulation

$$\varphi_{LB'_T}(p) > 0 \quad \forall p \in \text{lub}'(\mathcal{F}) \quad (17)$$

with the optimal value $\varphi_{LB'_T}(p)$ of (11). Hence, for each of the potentially many elements of $\text{lub}'(\mathcal{F})$ one convex optimization problem needs to be solved.

In the following we will successively construct polyhedral lower bounding sets in the node problems by cuts, which possess the potential to significantly reduce the number of these convex optimization problems to be solved. To this end we introduce polyhedral sets LB'_{PT} with $LB'_{PT} + \mathbb{R}_+^m =: \{y \in \mathbb{R}^m \mid y \geq \tilde{a}'_T, Ay \geq b\} \supseteq LB'_T + \mathbb{R}_+^m$ (cf. Figure 2). The main advantage of using this set is that the discarding condition (16) does not need to be checked by its equivalent reformulation (17), but one may resort to the sufficient condition

$$\text{lub}'(\mathcal{F}) \cap (LB'_{PT} + \mathbb{R}_+^m) = \emptyset$$

which may be rewritten as

$$\text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\} = \emptyset. \quad (18)$$

As the source of the system of inequalities $Ay \geq b$ we propose two types of cuts suggested in [11], where the second type of cuts is based on the first type and improves them under additional assumptions explained below. Initially the matrix A and the vector b are empty, resulting in $\text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\} = \text{lub}'(\mathcal{F})$ and $LB'_{PT} = \tilde{a}'_T + \mathbb{R}_+^m$. The general refinement step for $LB'_{PT} + \mathbb{R}_+^m = \{y \in \mathbb{R}^m \mid y \geq \tilde{a}'_T, Ay \geq b\}$ proceeds as follows.

$$\textit{Case 1: } \text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\} = \emptyset$$

Since (18) is sufficient for (16), X' can be discarded in view of Theorem 2.2.

$$\textit{Case 2: } \text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\} \neq \emptyset$$

Choose some $\bar{p} \in \text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\}$ (with a strategy presented below) and compute $\varphi_{LB'_T}(\bar{p})$ as the optimal value of the convex problem (11).

$$\textit{Case 2.1: } \varphi_{LB'_T}(\bar{p}) > 0$$

Then $\bar{p} \notin LB'_T + \mathbb{R}_+^m$ holds, and X' could be discarded if also the remaining $p \in \text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\}$ satisfied $p \notin LB'_T + \mathbb{R}_+^m$. To check the latter by polyhedral relaxation, we refine $LB'_{PT} + \mathbb{R}_+^m$ by information obtained during the computation of $\varphi_{LB'_T}(\bar{p})$, namely the cut $\bar{\lambda}^\top y \geq \bar{\lambda}^\top \bar{z}$, where (\bar{x}, \bar{t}) is an optimal point of (11), $\bar{z} := \bar{p} + \bar{t}e$, and $\bar{\lambda} \geq 0$ is a corresponding multiplier vector of the system of inequalities $f_T(x) \leq p + te$ [27,35]. The matrix A is then augmented by the row vector $\bar{\lambda}^\top$, the vector b by the entry $\bar{\lambda}^\top \bar{z}$, and the condition from the above Case 1 is checked again. In Section 6 we will refer to Algorithm 1 with the cuts from Case 2.1 as MOMIBB-c1.

Before we move on to Case 2.2, we briefly illustrate the construction from Case 2.1. In the situation of Figure 2 the set $\text{lub}'(\mathcal{F}) \cap (\tilde{a}'_T + \mathbb{R}_+^2)$ contains two elements, and \bar{p} is chosen to be the one to the right. Solving (11) (with the arrow symbolizing the vector e) yields $\varphi_{LB'_T}(\bar{p}) > 0$. With a corresponding multiplier vector $\bar{\lambda}$ a cut is added to update $LB'_{PT} + \mathbb{R}_+^2$ from the set $\tilde{a}'_T + \mathbb{R}_+^2$ to the set depicted with bold boundary lines. Solving (11) also for the second element in $\text{lub}'(\mathcal{F}) \cap (\tilde{a}'_T + \mathbb{R}_+^2)$ is now superfluous, since this point violates the cut. The subbox X' can thus be discarded.

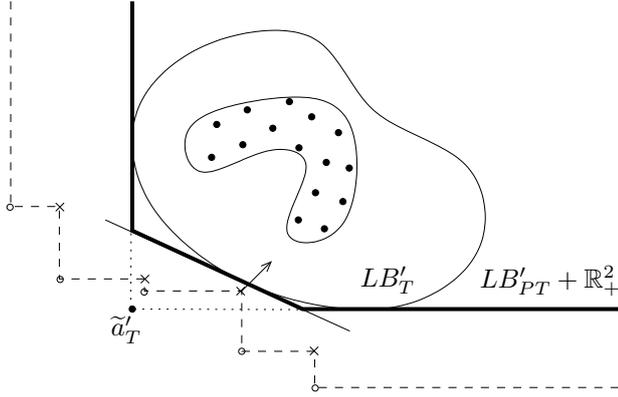


Fig. 2: A set $LB'_{PT} + \mathbb{R}_+^m$.

Case 2.2: $\varphi_{LB'_T}(\bar{p}) \leq 0$

In this case it is not possible to discard X' on the grounds of Theorem 2.2. However, under some algorithmic effort we may strengthen the above inequality $\bar{\lambda}^\top y \geq \bar{\lambda}^\top \bar{z}$ to a second type of cut by increasing its right-hand side while it stays valid for the *non-relaxed* mixed-integer partial upper image set $f(M(X')) + \mathbb{R}_+^m$. This can be realized by computing an optimal point \tilde{x} of the mixed-integer problem

$$\min_x \bar{\lambda}^\top f(x) \quad \text{s.t.} \quad x \in M(X') \quad (19)$$

and define the stronger valid inequality $\bar{\lambda}^\top y \geq \bar{\lambda}^\top f(\tilde{x})$ for $f(M(X')) + \mathbb{R}_+^m$.

Case 2.2.1: $\bar{\lambda}^\top \bar{p} < \bar{\lambda}^\top f(\tilde{x})$

This relation means that the valid inequality serves as a cut. The matrix A is augmented by the row vector $\bar{\lambda}^\top$, the vector b by the entry $\bar{\lambda}^\top f(\tilde{x})$, and the condition from the above Case 1 is checked again. In Section 6 we will refer to Algorithm 1 with the cuts from Case 2.2.1 as MOMIBB-c2. Note that for these cuts the condition $LB'_{PT} + \mathbb{R}_+^m \supseteq LB'_T + \mathbb{R}_+^m$ may no longer hold. Nevertheless, $p \notin LB'_{PT} + \mathbb{R}_+^m$ is still sufficient for $p \notin f(M(X')) + \mathbb{R}_+^m$. Hence the box X' can be discarded in case this holds for all other $p \in \text{lub}'(\mathcal{F})$ as well.

Case 2.2.2: $\bar{\lambda}^\top \bar{p} \geq \bar{\lambda}^\top f(\tilde{x})$

In this case the box X' cannot be discarded, and Algorithm 1 moves on.

Since generally the solution of the mixed-integer problem (19) is not possible in an efficient manner, the second type of cuts from Case 2.2 may only be beneficial under additional structural assumptions, like currently for quadratic functions. For more details on the two types of cuts we refer to [11].

In our numerical tests of these cutting strategies in Section 6 we proceed as follows to choose the point $\bar{p} \in \text{lub}'(\mathcal{F}) \cap \{y \in \mathbb{R}^m \mid Ay \geq b\}$ in Case 2. We rewrite $\{y \in \mathbb{R}^m \mid y \geq \tilde{a}'_T, Ay \geq b\} = \{y \in \mathbb{R}^m \mid \lambda^\top y \geq \lambda^\top z \text{ for all } (\lambda, z) \in \Omega\}$ with an appropriate finite set of parameters $\Omega \subseteq \mathbb{R}^m \times \mathbb{R}^m$ and put

$$\sigma(p) := \min\{\lambda^\top p - \lambda^\top z \mid (\lambda, z) \in \Omega\}.$$

Then the condition in Case 1 is equivalent to $\max_{p \in \text{lub}'(\mathcal{F})} \sigma(p) < 0$. Consequently in Case 2 we have $\max_{p \in \text{lub}'(\mathcal{F})} \sigma(p) \geq 0$, and the announced strategy is to choose any $\bar{p} \in \text{lub}'(\mathcal{F})$ at which this maximum is attained. In addition to being a formally natural choice, this also allows the following geometrical interpretation. Due to $\sigma(\bar{p}) \geq 0$ the inequalities $\lambda^\top \bar{p} - \lambda^\top z \geq 0$ are satisfied for all $(\lambda, z) \in \Omega$. If the vectors λ are chosen to be normalized, then the term $\lambda^\top \bar{p} - \lambda^\top z$ coincides with the geometrical distance of \bar{p} to the hyperplane $\{y \in \mathbb{R}^m \mid \lambda^\top y - \lambda^\top z = 0\}$. Therefore, among the $p \in \text{lub}'(\mathcal{F})$ with $\sigma(p) \geq 0$, the chosen \bar{p} maximizes the overall distance to all hyperplanes describing the boundary of $\{y \in \mathbb{R}^m \mid Ay \geq b\}$. This can be expected to lead to a cut with the potential to remove a significant portion of the remaining local upper bounds p .

6 Numerical Results

In the following, we present our numerical results for selected test instances including general mixed-integer nonconvex as well as mixed-integer convex quadratic instances. While for the former instances no algorithm with a performance guarantee was available so far, the latter are used to compare our new algorithm with those presented in [11] and [17].

All numerical tests were performed using MATLAB R2021a using a machine with Intel Core i9-10920X processor and 32GB of RAM. For this configuration 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. As mentioned in the MATLAB documentation [29], these results are highly version specific.

The termination tolerance for all instances was set to $\varepsilon = 0.1$. All single-objective mixed-integer subproblems were solved using Gurobi [22]. For single-objective continuous convex subproblems `fmincon` was used. This is also the configuration that was used in [11,18] for the numerical experiments to which we compare our algorithm later for the special case of convex problems. For all algorithms a time limit of 3600 seconds was set.

6.1 Performance of MOMIBB for the Nonconvex Case

Our algorithm is the first deterministic solver for multiobjective mixed-integer nonconvex optimization problems. Hence, we start our numerical tests considering such optimization problems and demonstrating the capabilities of Algorithm 1.

Within this subsection, we make use of convex continuous relaxations employing the α BB method [1,2] for convexification. As mentioned in the previous sections, we refer to this realization of Algorithm 1 (without any cutting strategies) as MOMIBB-c0. If, in addition, the first type of cuts from Section 5 (Case 2.1) is used, we refer to the algorithm as MOMIBB-c1. Finally, we denote by MOMIBB-c2 the realization of Algorithm 1 applying also the second type of cuts from Section 5 (Case 2.2.1).

First, we consider the biobjective mixed-integer nonconvex non-quadratic test instance

$$\min_x (x_1 + x_2 + x_5, x_3 + x_4 - \exp(x_5))^\top \quad \text{s.t.} \quad \begin{aligned} & \sum_{i=1}^4 x_i^2 \geq 1, \\ & x_1, x_2, x_3, x_4 \in [0, 1], \\ & x_5 \in [-4, 1] \cap \mathbb{Z} \end{aligned} \quad (\text{P1})$$

with quadratic but nonconvex constraints.

Without using any cutting strategies, MOMIBB-c0 computes an enclosure of the nondominated set of (P1) within 236.60 seconds. Using MOMIBB-c1, i.e., employing additional cuts, the computation time decreases only slightly to 226.59 seconds. The reason for this is that for (P1) the introduction of cuts did not allow to discard a significant amount of boxes earlier or to reduce the amount of problems (11) that needed to be solved. This changes in the forthcoming test instance (P2) where the use of cuts indeed reduces the computation time.

However, it is important to mention that it is always a recommendable strategy to use MOMIBB-c1 in favor of MOMIBB-c0. In fact, while the cuts introduced in MOMIBB-c1 may, in the worst case, reduce neither the total number of boxes that need to be considered nor the number of optimization problems (11) that need to be solved, the optimization problems (11) to obtain the cuts in MOMIBB-c1 are solved in MOMIBB-c0 anyway. In particular, MOMIBB-c1 never involves solving more subproblems than MOMIBB-c0.

Note that employing additional cuts, i.e., using MOMIBB-c2, is not a practical approach for this test instance since this would involve solving the (single-objective) mixed-integer nonconvex non-quadratic problems (19). This is (by now) not possible even using such advanced solvers as Gurobi.

Recall that the proposed algorithm MOMIBB works for an arbitrary number m of objectives. Thus, we continue this section with the triobjective nonconvex non-quadratic example

$$\min_x (x_1 + x_4, x_2 - x_4, x_3 - \exp(x_4) - 3)^\top \quad \text{s.t.} \quad \begin{aligned} & x_1^2 + x_2^2 \leq 1, \\ & \exp(x_3) \leq 1, \\ & x_1 x_2 (1 - x_3) \leq 1, \\ & x \in [-2, 2]^4, \\ & x_4 \in \mathbb{Z}. \end{aligned} \quad (\text{P2})$$

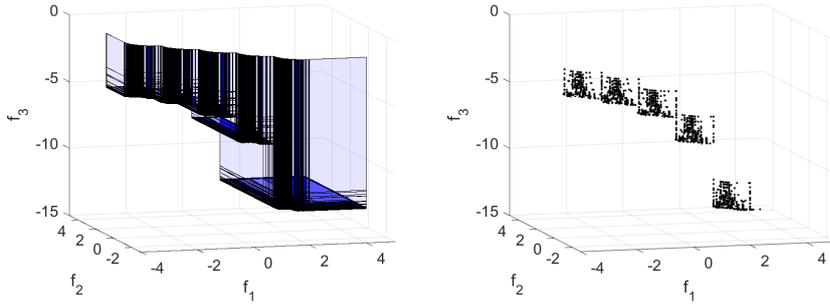


Fig. 3: Enclosure (left) and provisional nondominated set \mathcal{F} (right) for (P2) computed by MOMIBB-c0

For MOMIBB-c0 it took 528.34 seconds to compute an enclosure of the non-dominated set. In Figure 3 the enclosure as well as the provisional nondominated set \mathcal{F} are shown. By introducing cuts with MOMIBB-c1, the computation time can be reduced to 464.05 seconds. Thus, this example clearly shows the benefit of making use of the cutting strategies from Section 5. This decrease in computation time is mainly a result of the reduced number of subproblems that needed to be solved in order to discard certain boxes. More precisely, the computation time for `fmincon` reduced from 329.85 seconds for MOMIBB-c0 to 268.63 seconds for MOMIBB-c1. Again, using MOMIBB-c2 was not a practical approach for this test instance due to the nature of the subproblems (19).

To show the abilities of MOMIBB-c2 and MOMIBB_{direct}, we consider the scalable biobjective mixed-integer nonconvex but quadratic test instance

$$\begin{aligned}
 \min_x & \left(\sum_{i=1}^{k/2} x_i + \sum_{i=k+1}^{k+l/2} x_i, \sum_{i=k/2+1}^k x_i + \sum_{i=k+l/2+1}^n x_i \right)^\top \\
 \text{s.t.} & \sum_{i=1}^k x_i^2 \geq 1, \\
 & \sum_{i=k+1}^n x_i^2 \leq 9, \\
 & x_i \in [0, 1], \quad i = 1, \dots, k, \\
 & x_i \in [-3, 3] \cap \mathbb{Z}, \quad i = k+1, \dots, n.
 \end{aligned} \tag{P3}$$

with an even number $k \in \mathbb{N}$ of continuous variables, an even number $l \in \mathbb{N}$ of integer variables, and $n = k + l$ variables in total.

As explained in Section 4.3, modern solvers like Gurobi are able to handle even mixed-integer nonconvex quadratic optimization problems very well. This enables us to make use of advanced cutting strategies, i.e., using MOMIBB-c2, as well as completely omitting continuous convex relaxations and using $T = PIS$ which leads to MOMIBB_{direct}, see Section 4.3. Table 1 shows the numerical results for all those different strategies and for various choices of the

k	l	MOMIBB			MOMIBB _{direct}
		c0	c1	c2	
2	2	34.76	34.89	32.42	14.39
2	4	165.39	157.80	146.25	70.15
2	6	409.23	373.19	290.06	81.46
4	2	247.33	243.84	180.62	100.24
4	4	1322.00	1231.51	931.07	536.80
4	6	1764.09	1664.65	1291.60	455.38
6	2	1240.25	1094.80	907.16	470.45
6	4	-	-	3144.38	2421.86

Table 1: Comparison of computation times for (P3) in seconds

numbers $k, l \in \mathbb{N}$ of continuous and integer variables. A ‘-’ indicates that no solution satisfying the termination criterion was found within the time limit of 3600 seconds.

Introducing the first kind of cuts, i.e., using MOMIBB-c1, in general, slightly decreases the computation times in comparison to MOMIBB-c0. This is basically the same behavior we have seen for (P1) before. Additionally introducing the second type of cuts obtained by solving the mixed-integer nonconvex quadratic problem (19), i.e., using MOMIBB-c2, reduces the computation times more significantly. The reason for this is mainly that the number of boxes in the pre-image space that need to be considered decreases noticeably for MOMIBB-c2. For example, regarding the instance of (P3) with $k = l = 2$, the number of boxes (in comparison to MOMIBB-c0) decreases by a third using MOMIBB-c2 (66 instead of 99) while it stays the same when using MOMIBB-c1.

As a final strategy, we use MOMIBB_{direct}, i.e., we completely omit the continuous convex relaxation and choose $T = PIS$ instead, as discussed in Section 4.3. This strategy massively reduces the overall computation time, making it more than twice as fast as using continuous convex relaxation. This demonstrates how for specific types of optimization problems, our algorithm benefits from the advances in corresponding state-of-the-art solvers. In Figure 4 the results, i.e., the computed enclosure of the nondominated set of (P3) and the provisional nondominated set \mathcal{F} , for (P3) with $k = l = 2$ and MOMIBB_{direct} are shown.

6.2 Comparison with Other Algorithms for the Convex Case

While our algorithm is the first of its kind to solve multiobjective mixed-integer nonconvex optimization problems, several solvers for the convex case have already been proposed in the literature. In this section, we compare our algorithm MOMIBB to two of them, namely HyPaD [17] and MOMIX [11].

Since HyPaD also computes an enclosure, the termination criterion of HyPaD matches the termination criterion of our algorithm. This allows for a fair comparison. While our algorithm and HyPaD both work with the same image

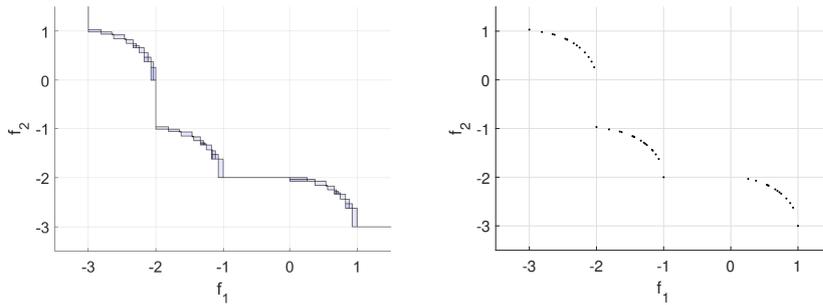


Fig. 4: Enclosure (left) and provisional nondominated set \mathcal{F} (right) for (P3) with $k = 2$ continuous and $n - k = 2$ integer variables computed by $\text{MOMIBB}_{\text{direct}}$

space based termination criterion (depending on the parameter ε), MOMIX uses a criterion in the pre-image space (depending on δ). All results for the MOMIX algorithm from [11] were computed using a choice of $\delta = 0.1$ as termination criterion. Hence, quantitative comparisons are only possible to some extent and we will focus mostly on qualitative comparisons.

The test instance

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

is taken from [11]. It is scalable in both the number $k \in \mathbb{N}$ of continuous and $l \in \mathbb{N}$ of integer variables, where the number of continuous variables needs to be even and $n = k + l$.

Since this test instance is quadratic, we can employ all solution strategies MOMIBB-c0 , MOMIBB-c1 , MOMIBB-c2 , and $\text{MOMIBB}_{\text{direct}}$ as in the previous test instance (P3). The results for all configurations of our algorithm as well as for MOMIX and HyPaD are shown in Table 2, where ‘-’ indicates that no solution satisfying the termination criterion was found within the time limit of 3600 seconds. For MOMIX two different branching strategies can be used. We present here only the best result (i.e., the shortest computation time) of all of them.

For all instances with $k = 2$ continuous variables, the results for $\text{MOMIBB}_{\text{direct}}$ are significantly better than those obtained for MOMIBB-c0 , MOMIBB-c1 , and MOMIBB-c2 . We have already seen this behavior for test instance (P3). For $k = 4$, $l = 1$ $\text{MOMIBB}_{\text{direct}}$ was still better than MOMIBB-c0 . However, it no longer outperformed MOMIBB-c1 and MOMIBB-c2 .

k	l	MOMIBB			MOMIBB _{direct}	MOMIX	HyPaD
		c0	c1	c2			
2	1	33.31	20.35	25.03	5.22	22.65	1.48
2	2	185.19	104.52	116.36	27.15	129.13	3.37
2	3	1774.12	586.21	644.45	180.25	752.27	5.53
4	1	694.54	209.82	229.03	268.06	1318.00	1.87
4	10	-	-	-	-	-	30.96

Table 2: Comparison of computation times for (T4) in seconds

It is also in line with the previous examples that MOMIBB-c1 performed better than MOMIBB-c0. Again, this is no surprise since MOMIBB-c1 can never perform worse than MOMIBB-c0 as it only makes more use of the data already obtained. MOMIBB-c2 was not able to yield a better performance than MOMIBB-c1 for this test instance.

For all instances MOMIBB-c1 as well as MOMIBB_{direct} performed better than MOMIX. In particular, comparing the results for MOMIBB_{direct} and MOMIX, the advantage of MOMIBB_{direct} seems to be quite stable (roughly factor 5). Note that since both, our algorithm and MOMIX, are branch-and-bound approaches and both, MOMIBB_{direct} and MOMIX, make use of the ability of Gurobi to solve single-objective mixed-integer quadratic optimization problems, this is also the most meaningful comparison quality-wise.

HyPaD on the other hand operates almost entirely in the image space. Thus, it is no surprise that it is the fastest of all tested algorithms and is able to solve even larger instances. In fact, it was demonstrated in [18] that HyPaD is able to solve instances with up to $l = 30$ integer variables or $k = 200$ continuous variables. Clearly, for such dimensions, MOMIBB with a branching in the pre-image space will have difficulties. On the other hand, this branching in the pre-image space is the reason why nonconvex functions can be handled by employing convexification using for instance α BB underestimators, while HyPaD cannot be applied in this case. Hence, unless it is inevitable to use pre-image space techniques such as branch-and-bound, for example to handle nonconvexity, it is beneficial to stick to such algorithms as HyPaD that work almost entirely in the criterion space.

In case scalarizations of *MOMIP* can be solved fast and reliably, one could make use of algorithms as the ones from [19,36] that also generate an approximation of the nondominated set.

Statements and Declarations

The work of the first and the third author is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within Project-ID 432218631. The authors have no competing interests to declare that are relevant to the content of this article. Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. Claire S Adjiman, Ioannis P Androulakis, and Christodoulos A Floudas. A global optimization method, α BB, for general twice-differentiable constrained NLPs – II. Implementation and computational results. *Computers & Chemical Engineering*, 22(9):1159–1179, 1998.
2. Claire S Adjiman, Stefan Dallwig, Christodoulos A Floudas, and Arnold Neumaier. A global optimization method, α BB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.
3. Pietro Belotti. Disjunctive cuts for nonconvex MINLP. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 117–144. Springer Science & Business Media, 2012.
4. Timo Berthold. *Heuristic Algorithms in Global MINLP Solvers*. Verlag Dr. Hut, 2015.
5. Regina Burachik, C. Yalcin Kaya, and Mohammed Mustafa Rizvi. Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. *Engineering Optimization*, 54(8):1413–1425, 2021.
6. Samuel Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2, Ser. A):479–495, 2009.
7. Samuel Burer and Adam Letchford. Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Operations Research and Management Science*, 17:97–106, 2012.
8. Guillermo Cabrera-Guerrero, Matthias Ehrgott, Andrew Mason, and Andrea Raith. Bi-objective optimisation over a set of convex sub-problems. *Annals of Operations Research*, DOI: 10.1007/s10479-020-03910-3, 2021.
9. Guang-ya Chen, Xuexiang Huang, and Xiaogi Yang. *Vector Optimization: Set-valued and Variational Analysis*. Springer Science & Business Media, 2006.
10. Kerstin Dächert, Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research*, 260(3):841–855, 2017.
11. Marianna De Santis, Gabriele Eichfelder, Julia Niebling, and Stefan Rocktäschel. Solving multiobjective mixed integer convex optimization problems. *SIAM Journal on Optimization*, 30(4):3122–3145, 2020.
12. Erik Diessel. An adaptive patch approximation algorithm for bicriteria convex mixed-integer problems. *Optimization*, DOI: 10.1080/02331934.2021.1939699, 2021.
13. Matthias Ehrgott. *Multicriteria Optimization*. Springer Science & Business Media, 2005.
14. Gabriele Eichfelder and Patrick Groetzner. A note on completely positive relaxations of quadratic problems in a multiobjective framework. *Journal of Global Optimization*, 82:615–626, 2022.
15. Gabriele Eichfelder, Peter Kirst, Laura Meng, and Oliver Stein. A general branch-and-bound framework for continuous global multiobjective optimization. *Journal of Global Optimization*, 80(1):195–227, 2021.
16. Gabriele Eichfelder and Oliver Stein. Limit sets in continuous global multiobjective optimization. *Optimization*, DOI: 10.1080/02331934.2022.2092479, 2022.
17. Gabriele Eichfelder and Leo Warnow. A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. Preprint 2021-08-8541, *Optimization Online*, 2021.
18. Gabriele Eichfelder and Leo Warnow. On implementation details and numerical experiments for the hypad algorithm to solve multi-objective mixed-integer convex optimization problems. Preprint 2021-08-8538, *Optimization Online*, 2021.
19. Gabriele Eichfelder and Leo Warnow. An approximation algorithm for multi-objective optimization problems using a box-coverage. *Journal of Global Optimization*, DOI: 10.1080/s10898-021-01109-9, 2021.
20. Jose Fernandez and Boglarka Toth. Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods. *Computational Optimization and Applications*, 42(3):393–419, 2009.
21. Alfred Göpfert, Hassan Riahi, Christiane Tammer, and Constantin Zalinescu. *Variational Methods in Partially Ordered Spaces*. Springer Science & Business Media, 2006.
22. LLC Gurobi Optimization. *Gurobi Reference Manual*. 2020.

23. Eldon Hansen and G William Walster. *Global Optimization Using Interval Analysis: Revised and Expanded*. CRC Press, 2003.
24. Gokhan Kirlik and Serpil Sayin. Bilevel programming for generating discrete representations in multiobjective optimization. *Mathematical Programming*, 169:585–604, 2018.
25. Peter Kirst, Oliver Stein, and Paul Steuermann. Deterministic upper bounds for spatial branch-and-bound methods in global minimization with nonconvex constraints. *TOP*, 23(2):591–616, 2015.
26. Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. On the representation of the search region in multi-objective optimization. *European Journal of Operational Research*, 245(3):767–778, 2015.
27. Andreas Löhne, Birgit Rudloff, and Firdevs Ulus. Primal and dual approximation algorithms for convex vector optimization problems. *Journal of Global Optimization*, 60(4):713–736, 2014.
28. Pierre Loridan. ε -solutions in vector minimization problems. *Journal of Optimization Theory and Applications*, 43(2):265–276, 1984.
29. MATLAB. Matlab Bench Documentation. <https://www.mathworks.com/help/matlab/ref/bench.html>. Accessed 2022-01-10.
30. Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer Science & Business Media, 2012.
31. Arnold Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.
32. Christoph Neumann and Oliver Stein. Generating feasible points for mixed-integer convex optimization problems by inner parallel cuts. *SIAM Journal on Optimization*, 31(3):2396–2428, 2021.
33. Christoph Neumann, Oliver Stein, and Nathan Sudermann-Merx. A feasible rounding approach for mixed-integer optimization problems. *Computational Optimization and Applications*, 72(2):309–337, 2019.
34. Christoph Neumann, Oliver Stein, and Nathan Sudermann-Merx. Granularity in nonlinear mixed-integer optimization. *Journal of Optimization Theory and Applications*, 184(2):433–465, 2020.
35. Julia Niebling and Gabriele Eichfelder. A branch-and-bound-based algorithm for nonconvex multiobjective optimization. *SIAM Journal on Optimization*, 29(1):794–821, 2019.
36. Soghra Nobakhtian and Narjes Shafiei. A Benson type algorithm for nonconvex multiobjective programming problems. *TOP*, 25:271–287, 2017.
37. Tyler Perini, Natasha Boland, Diego Pecin, and Martin Savelsbergh. A criterion space method for biobjective mixed integer programming: The boxed line method. *INFORMS Journal on Computing*, 32:16–39, 2019.
38. Anthony Przybylski and Xavier Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3):856–872, 2017.
39. Anthony Przybylski, Kathrin Klamroth, and Renaud Lacour. A simple and efficient dichotomic search algorithm for multi-objective mixed integer linear programs. Preprint 1911.08937, arXiv, 2019.
40. Siegfried R Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.
41. Stefan Ruzika and Margaret M Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, 2005.
42. Edward MB Smith and Constantinos C Pantelides. Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 21:791–796, 1997.
43. Edward MB Smith and Constantinos C Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 23(4-5):457–478, 1999.
44. Mohit Tawarmalani and Nikolaos V Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Springer Science & Business Media, 2013.