Journal of Global Optimization
https://doi.org/10.1007/s10898-024-01440-x

Using dual relaxations in multiobjective mixed-integer convex quadratic programming

Marianna De Santis[1] · Gabriele Eichfelder[2] · Daniele Patria[3] · Leo Warnow[2]

Received: 11 December 2023 / Accepted: 3 October 2024

Abstract
We present a branch-and-bound method for multiobjective mixed-integer convex quadratic programs that computes a superset of efficient integer assignments and a coverage of the nondominated set. The method relies on outer approximations of the upper image set of continuous relaxations. These outer approximations are obtained addressing the dual formulations of specific subproblems where the values of certain integer variables are fixed. The devised pruning conditions and a tailored preprocessing phase allow a fast enumeration of the nodes. Despite we do not require any boundedness of the feasible set, we are able to prove that the method stops after having explored a finite number of nodes. Numerical experiments on a broad set of instances with two, three, and four objectives are presented.

Keywords Multiobjective optimization · Convex quadratic optimization · Mixed-integer quadratic programming · Branch-and-bound algorithm

Mathematics Subject Classification 90C11 · 90C25 · 90C29 · 90C57

1 Introduction

The area of multiobjective mixed-integer programming (MOMIP) is receiving growing attention from the operations research and optimization community, both for its practical relevance and for the mathematical challenge of solving MOMIP problems. Applications can be found,

✉ Marianna De Santis
marianna.desantis@unifi.it

Gabriele Eichfelder
gabriele.eichfelder@tu-ilmenau.de

Daniele Patria
daniele.patria@uniroma1.it

Leo Warnow
leo.warnow@tu-ilmenau.de

1 Dipartimento di Ingegneria dell'informazione, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy

2 Institute of Mathematics, Technische Universität Ilmenau, Po 10 05 65, 98684 Ilmenau, Germany

3 Dipartimento di Ingegneria Informatica Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto, 25, 00185 Roma, Italy

for example, in transportation, design of water distribution networks, and biology [1–4]. When addressing multiobjective programming problems, a typical goal is to compute an approximation of the nondominated set, which corresponds to the set of optimal values. The proper definition of such an approximation is a debated topic (see [5] for an overview).

A possibility is given by the concept of an *enclosure*, exploited in several recent approaches [6, 7] and in the mixed-integer context, too [8, 9]. Loosely speaking, an enclosure is a well-structured set in the image space, as for example a union of boxes, which contains the nondominated set as a subset. Using the enclosure concept, we have a termination criterion for global algorithms: a multiobjective programming problem can be considered solved to a certain precision as soon as the quality of the enclosure is below a specific value. We will give the formal definition of an enclosure in Sect. 2.

Various methods with correctness guarantees proposed in the literature are branch-and-bound frameworks. This includes, for instance, [10–12] for multiobjective integer programming and [8, 13, 14] for MOMIP. For a broader survey of branch-and-bound methods, mainly for linear MOMIP problems, we refer to [15]. The survey in [16] extends this collection by also including approaches that do not use a branch-and-bound framework.

In this paper, we develop a branch-and-bound method that, using dual relaxations as a key ingredient, is guaranteed to compute an enclosure for the nondominated set of a multiobjective mixed-integer convex quadratic programming problem in a finite number of iterations. Differently from approaches that work exclusively in the image space, our algorithm is also able to deliver a superset of the set of efficient integer assignments, that in turn is needed as input for some existing approaches (see, e.g., [17]). Here, an efficient integer assignment is a fixing of the integer variables in such a way that there exists an efficient point of the problem with exactly this fixing. Note that for multiobjective mixed-integer problems, it can happen that there is a large number of efficient integer assignments. In fact, there exist instances of such optimization problems that cannot be solved without a full enumeration of all integer assignments. The reason for that is that it can happen that all integer assignments contribute to different parts of the nondominated set. This clearly represents a big difference with respect to the single-objective case and at the same time a big challenge regarding the development of solution algorithms for MOMIPs. In the definition of branch-and-bound methods, this emphasizes the need for strategies for a fast enumeration of the nodes.

Starting from some of the ideas presented in [18] to deal with purely integer unconstrained problems, in this paper, we address the difficulties of handling the presence of both continuous and integer variables and linear inequality constraints. In contrast to the purely integer case, the nondominated set of the corresponding multiobjective mixed-integer nonlinear optimization problems is an, in general, infinite set and cannot be computed exactly. Consequently, solving such optimization problems usually refers to computing an approximation of the nondominated set, the efficient set, or both of them. Thus, different solution techniques are needed. The method that we present in this paper computes both a coverage of the nondominated set and a superset of the set of efficient integer assignments.

More formally, we focus on optimization problems where the goal is to minimize $m \geq 2$ quadratic objective functions given by $f_j : \mathbb{R}^n \to \mathbb{R}$,

$$f_j(x) = x^\top Q_j x + (c^j)^\top x + a_j,$$

for all $j \in [m] := \{1, \ldots, m\}$ with symmetric positive definite matrices $Q_1, \ldots, Q_m \in \mathcal{S}^n$, vectors $c^1, \ldots, c^m \in \mathbb{R}^n$, and scalars $a_1, \ldots, a_m \in \mathbb{R}$. We recall that under the assumption that all the matrices Q_j are positive definite, we have that all the functions f_j are strongly convex. This means that there exist some $\nu > 0$ such that for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0, 1]$ it holds

$$\nu\lambda(1-\lambda)\|x-y\|_2^2 + f_j(\lambda x + (1-\lambda)\, y) \le \lambda\, f_j(x) + (1-\lambda)\, f_j(y)$$

for $j \in [m]$, where with $\|\cdot\|_2$ we denote the Euclidean norm. This also implies the strict convexity of the functions, i.e., for any $x, y \in \mathbb{R}^n$ with $x \neq y$ and any $\lambda \in (0, 1)$ we have $f_j(\lambda x + (1-\lambda)\, y) < \lambda\, f_j(x) + (1-\lambda)\, f_j(y)$ for $j \in [m]$. These convexity assumptions will be essential to derive the finiteness result for our new algorithm.

Assuming that the objective functions are quadratic allows us to perform many of the expensive calculations in a preprocessing phase. What is more important, we can make use of simple dual formulations within our procedure which are a main aspect to make our algorithm fast and efficient. Thus, in summary, compared to the algorithm presented in [18], we extend the theory and the algorithm to mixed-integer problems (compared to the pure integer case in [18]), to linearly constrained problems (compared to unconstrained problems in [18]), and we additionally speed up the algorithm by making use of dual formulations.

The multiobjective mixed-integer quadratic programming problem which we study in this paper is given as

$$\begin{array}{ll} \min\limits_{x} & (f_1(x), \ldots, f_m(x))^\top \\ \text{s.t.} & Ax \le b \\ & x_i \in \mathbb{Z} \text{ for all } i \in [k] \\ & x \in \mathbb{R}^n \end{array} \tag{MOMIQP}$$

with a matrix $A \in \mathbb{R}^{p \times n}$, a vector $b \in \mathbb{R}^p$, and with $1 \le k \le n$, i.e., we assume that at least one variable can attain integer values only. We do not need any assumption on the boundedness of the feasible set. In particular, we are not assuming any lower or upper bounds on the variables, i.e., no box constraints, as it is often required, for instance in [14] or for branch-and-bound based methods with a partitioning of the starting box in the pre-image space as in [8, 13]. In the following, the feasible set of (MOMIQP) is denoted by S, i.e.,

$$S := \{x \in \mathbb{Z}^k \times \mathbb{R}^{n-k} \mid Ax \le b\}.$$

Note that for $k = n$ we have the special case of a multiobjective integer quadratic programming problem, for which our method will be exact. This means that it will be able to detect the whole nondominated set, which is a finite set.

The paper is organized as follows. In Sect. 2, we give some standard definitions for multiobjective optimization and we formally recall what an enclosure of the nondominated set is. In Sect. 3 we define the subproblems that we address at the nodes of our branch-and-bound algorithm according to our branching strategy that works by fixing integer variables. In Sect. 4 we present the theoretical results that allow to avoid an infinite enumeration of nodes even in case problem (MOMIQP) has an unbounded feasible set. In Sect. 5 a scheme of our branch-and-bound algorithm is presented and in Sect. 6, we see how the dual relaxations may come into play to save computational effort. Eventually, in Sect. 7, numerical results are reported and in Sect. 8 we draw some conclusions.

2 Basic notions and definitions

For the following notions as well as an introduction to multiobjective optimization we refer, for instance, to [19]. We use the standard optimality notion based on the componentwise partial ordering in the image space. A point $\bar{x} \in S$ is called an efficient point for (MOMIQP) if there is no feasible point $x \in S$ with $f(x) \neq f(\bar{x})$ and with $f(x) \leq f(\bar{x})$. Here and in the following, $\leq$ and $<$ are understood componentwise. The image $f(\bar{x})$ of an efficient point for (MOMIQP) is called nondominated, and the image set of all efficient points is denoted as the nondominated set $\mathcal{N}$ (also known, specifically for $m = 2$, as Pareto front). Thanks to Corollary 8.1 in the Appendix, we have that the nondominated set of a MOMIP problem with strongly convex objective functions is a bounded set. In particular, due to our assumptions, this holds for our problem (MOMIQP). Hence, it is guaranteed that a closed box

$$B := [z, Z] := \{x \in \mathbb{R}^m \mid z \leq x \leq Z\} \tag{1}$$

with $\mathcal{N} \subseteq \operatorname{int}(B) = (z, Z) := \{x \in \mathbb{R}^m \mid z < x < Z\}$, where $z, Z \in \mathbb{R}^m$, always exists. As already mentioned, we aim at approximating the nondominated set $\mathcal{N}$ by an *enclosure* that can be defined as follows (see [20]).

Definition 1 Let $\mathcal{L}, \mathcal{U} \subseteq \mathbb{R}^m$ be two finite sets with $\mathcal{N} \subseteq \mathcal{L} + \mathbb{R}^m_+$ and $\mathcal{N} \subseteq \mathcal{U} - \mathbb{R}^m_+$. Then $\mathcal{L}$ is called a lower bound set, $\mathcal{U}$ is called an upper bound set, and the set $\mathcal{A}$ which is given as

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

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

Note that for the elements of the set $\mathcal{U}$ one cannot take just objective function values $f(x)$ of feasible points $x \in S$, as one might be used to from single-objective global optimization. Instead we need another concept, for instance the one of so-called local upper bounds, which we will introduce below. A lower bound set $\mathcal{L}$ can be computed as a union of ideal points of certain subproblems of (MOMIQP) which we discuss in the forthcoming Section 3.

The quality of an enclosure $\mathcal{A}$ is given by its width $w(\mathcal{A})$. It is defined in [6] as the optimal value of

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

where $s(l, u) := \min \{u_i - l_i \mid i \in [m]\}$ denotes the shortest edge length of a box $[l, u]$. The surprising fact that this quality measure is based on the shortest edge length of the boxes, and not on the largest as typically expected in global optimization, is due to a desired relation to ε-optimality. For $\varepsilon > 0$ a point $\bar{x} \in S$ is called ε-efficient for (MOMIQP) if there exists no $x \in S$ with $f(x) \neq f(\bar{x}) - \varepsilon e$ and $f(x) \leq f(\bar{x}) - \varepsilon e$, where e represents the all-ones vector. We denote the image set of all ε-efficient points by $\mathcal{N}_\varepsilon$. According to Lemma 3.1 in [6], if $\mathcal{A}$ is an enclosure of $\mathcal{N}$ with $w(\mathcal{A}) < \varepsilon$ then any $x \in S$ with $f(x) \in \mathcal{A}$ is at least ε-efficient for (MOMIQP). In other words, $\mathcal{A} \cap f(S) \subseteq \mathcal{N}_\varepsilon$ holds. This is the natural extension of ε-optimality as used in single-objective global optimization. A more detailed discussion and extensive motivation for this quality measure is provided in [6, 7].

As already mentioned, and widely discussed in the literature, a proper concept to obtain an upper bound set $\mathcal{U}$ are the so-called local upper bounds which have been presented in [21]. In the following definition, we use the generalized definition of local upper bounds from [20, Def. 4.1] with the set B from (1) as the so-called initial area of interest. Within the definition we further make use of stable sets. These are sets $N \subseteq \mathbb{R}^m$ where for any two points $y^1, y^2 \in N$ with $y^1 \neq y^2$ it holds that $y^1 \nleq y^2$, i.e., all elements of N are pairwise non-comparable.

Definition 2 Let $N \subseteq \mathbb{R}^m$ be a finite and stable set. Then the lower search region for N is $s(N) := \mathrm{int}(B) \setminus (N + \mathbb{R}^m_+)$ and the lower search zone for some $u \in \mathbb{R}^m$ is $c(u) := \{y \in \mathrm{int}(B) \mid y < u\}$. A set $\mathcal{U} = U(N) \subseteq \mathbb{R}^m$ is called local upper bound set given N if $s(N) = \bigcup_{u \in U(N)} c(u)$ and if $\{u^1\} - \mathrm{int}(\mathbb{R}^m_+) \nsubseteq \{u^2\} - \mathrm{int}(\mathbb{R}^m_+)$ 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).

We remark that the set B within Definition 2 does not necessarily have to be a box as in (1), but can be chosen as an arbitrary subset of $\mathbb{R}^m$ with $\mathcal{N} \subseteq \mathrm{int}(B)$, see [20, Assumption 4.3].

3 Building subproblems by fixing variables

The algorithm we propose is a branch-and-bound method that works by fixing the values of certain integer variables. Our enumeration strategy is depth-first and we always branch by fixing the value of one of the k integer variables. A crucial property of our algorithm is that the set of fixed variables only depends on the depth of the node in the branch-and-bound tree. We thus lose the flexibility of choosing the best branching variable, but this strategy allows us to process a single node in the tree much faster. As it will be clarified later, this branching strategy, already successfully used in single-objective mixed-integer optimization [22–25], allows to perform a preprocessing phase for faster computations at the nodes. More precisely, at a generic level $d \in [k] \cup \{0\}$ of the branch-and-bound tree, the variables $x_1, \dots, x_d$ are fixed to certain integer values, say, $r_1, \dots, r_d \in \mathbb{Z}$. In particular, the order in which integer variables are fixed is predetermined. This means that we start by fixing the value of x_1 at level $d = 1$, continue by fixing the value of x_2 at level $d = 2$, and so on, until we fix the last integer variable x_k at level $d = k$. Hence, at every node of the branch-and-bound tree at the same level $d \in [k] \cup \{0\}$ the same set of integer variables is fixed to certain (different) values. We remark that at level $d = 0$ (root node) no variable is fixed and we have the original problem. The full algorithmic scheme of our method is reported in Sect. 5.

Let $r = (r_1, \dots, r_d)^\top \in \mathbb{Z}^d$ be a vector of integer fixings. This vector defines a specific node at level d of the branch-and-bound tree. For every $j \in [m]$, we define, as in [18, Lemma 3.1], the function $f_j^r : \mathbb{R}^{n-d} \to \mathbb{R}$ by $f_j^r(x) := f_j(r_1, \dots, r_d, x_1, \dots, x_{n-d})$. This function can be expressed explicitly as

$$f_j^r(x) = x^\top Q_j^d x + (c^{j,r})^\top x + a_{j,r},$$

where the positive definite symmetric matrix Q_j^d is obtained by deleting the corresponding d rows and columns of Q_j and $c^{j,r}$ and $a_{j,r}$ are set to

$$c_{i-d}^{j,r} := c_i^j + 2 \sum_{l=1}^{d} q_{li} r_l, \quad \text{for } i = d+1, \dots, n$$

and

$$a_{j,r} := a_j + \sum_{l=1}^{d} c_l r_l + \sum_{l=1}^{d} \sum_{i=1}^{d} q_{li} r_l r_i.$$

Similarly, we define the matrix $A^d \in \mathbb{R}^{p \times (n-d)}$ and the vector $b^r \in \mathbb{R}^p$ by taking into account the fixings, i.e., A^d denotes the matrix which is obtained from A by deleting the first d columns and $b^r := b - A(r_1, \dots, r_d, 0, \dots, 0)^\top$.

We consider the following continuous relaxation of (MOMIQP) induced by that fixing $r \in \mathbb{Z}^d$ of the first d integer variables:

$$\begin{aligned} \min_x \quad & (f_1^r(x), \ldots, f_m^r(x))^\top \\ \text{s.t.} \quad & A^d x \leq b^r \\ & x \in \mathbb{R}^{n-d}. \end{aligned} \tag{MOP^r}$$

In our method, we mainly use these continuous subproblems to compute a lower bound set $\mathcal{L}$ for an enclosure of the nondominated set of (MOMIQP) and to check whether the node corresponding to the fixing $r \in \mathbb{Z}^d$ of integer variables can be pruned. In fact, we do not consider (MOP^r) directly, but an outer approximation of the corresponding upper image set

$$\mathcal{P}^r := \{f^r(x) \in \mathbb{R}^m \mid A^d x \leq b^r,\ x \in \mathbb{R}^{n-d}\} + \mathbb{R}^m_+.$$

The simplest outer approximation is determined by the ideal point of this set, which is componentwise calculated as $\min\{y_j \in \mathbb{R} \mid y \in \mathcal{P}^r\}$ for $j \in [m]$. By our assumptions, these minima exist in case the feasible set $S^r := \{x \in \mathbb{R}^{n-d} \mid A^d x \leq b^r\}$ of (MOP^r) is nonempty. This approximation using the ideal point corresponds to an outer approximation derived by m supporting hyperplanes to the set $\mathcal{P}^r$ with normal vectors equal to the m unit vectors. As this outer approximation is very rough, we allow improved outer approximations. In this respect, let $L \subseteq \{y \in \mathbb{R}^m_+ \mid \|y\|_1 = 1\}$ be a finite set of nonnegative vectors which includes all m unit vectors. We decided here for the 1-norm but any other norm can also be used for normalizing the vectors. This set defines the hyperplanes which are used for the outer approximation of $\mathcal{P}^r$. The derived approximation of the upper image set $\mathcal{P}^r$ will be involved in our pruning condition which we define later.

In order to compute the outer approximation, at a node, we solve the $|L|$ continuous single-objective subproblems

$$\begin{aligned} \min_x \quad & \ell^\top f^r(x) \\ \text{s.t.} \quad & A^d x \leq b^r \\ & x \in \mathbb{R}^{n-d}. \end{aligned} \tag{$P^r(\ell)$}$$

In fact, we will examine the dual problems of these subproblems, see Sect. 6. In case problem ($P^r(\ell)$) is infeasible, i.e., in case we have

$$S^r = \{x \in \mathbb{R}^{n-d} \mid A^d x \leq b^r\} = \emptyset\,, \tag{Inf}$$

the node can be pruned, see the results in Sect. 4.1.

Otherwise, in case problem ($P^r(\ell)$) is feasible, we define $\varphi^r(\ell)$ to be its optimal value for $\ell \in L$. Moreover, we denote by $x^{*\ell,r} \in \mathbb{R}^{n-d}$ its unique minimizer, which exists due to the strong convexity of the objective function. Note that for $\ell = e^j$, the j-th unit vector, we minimize $\ell^\top f^r(\cdot) = f_j^r(\cdot)$. Hence, in that case $x^{*\ell,r}$ denotes the unique minimizer of f_j^r with respect to S^r. In particular, $\varphi^r(e^j) = f_j^r(x^{*e^j,r})$ gives the j-th component of the ideal point of the set $\mathcal{P}^r$.

Furthermore, in case of feasibility, i.e., in case $S^r \neq \emptyset$, we define

$$\alpha(d+1) := \min_{\ell \in L} x_1^{*\ell,r}, \qquad \beta(d+1) := \max_{\ell \in L} x_1^{*\ell,r}\,, \tag{2}$$

and the interval

$$[\,\lfloor \alpha(d+1) \rfloor\,,\ \lceil \beta(d+1) \rceil\,]\ . \tag{I}$$

Recall that the vector $r \in \mathbb{Z}^d$ of integer fixings corresponds to a node at level d of the branch-and-bound tree. In case $d < k$, the interval (I) basically defines the range of values $r'_{d+1} \in \mathbb{Z}$ for which, within our algorithm, child nodes with x_{d+1} fixed to $r'_{d+1} \in \mathbb{Z}$ need to be considered. More importantly, we will show in Lemma 4.8 that all child nodes corresponding to $r' := (r_1, \ldots, r_d, r'_{d+1}) \in \mathbb{Z}^{d+1}$ with r'_{d+1} outside that interval and far enough can safely be pruned. This is one of the key results that ensures finiteness of the overall algorithm.

In the following, we briefly explain how exactly the child nodes at level $d+1 \leq k$ corresponding to such vectors $r' \in \mathbb{Z}^{d+1}$ of integer fixings are explored within our algorithm. At the first child node, x_{d+1} is fixed to $r'_{d+1} = \lfloor \alpha(d+1) \rfloor$. Then its sibling nodes are computed by consecutively fixing x_{d+1} to increasing integer values $r'_{d+1} \in \{\lfloor \alpha(d+1) \rfloor + 1, \lfloor \alpha(d+1) \rfloor + 2, \ldots, \lceil \beta(d+1) \rceil\}$. The method continuous with fixing x_{d+1} to increasing integer values $r'_{d+1} > \lceil \beta(d+1) \rceil$ until it reaches the first assignment of r'_{d+1} for which the node corresponding to $r' \in \mathbb{Z}^{d+1}$ can be pruned by one of the conditions we present in the forthcoming Sect. 4. Since that implies that also all child nodes with even larger values of the integer variable x_{d+1} can be pruned, the algorithm continues by exploring those nodes corresponding to fixings of $r'_{d+1} < \lfloor \alpha(d+1) \rfloor$. Again, starting from $\lfloor \alpha(d+1) \rfloor - 1$, the value of r'_{d+1} is decreased until the first child node which can be pruned based on the results from Sect. 4 is found. The rules adopted to fix the variables are outsourced in Algorithm 1. Again, the full branch-and-bound algorithm is presented in Sect. 5.

Algorithm 1 Update r_d

INPUT: $r_d, \alpha(d)$
OUTPUT: r_d
1: **if** $r_d \geq \lfloor \alpha(d) \rfloor$ **then**
2: Set $r_d = r_d + 1$
3: **else**
4: Set $r_d = r_d - 1$
5: **end if**

To conclude this section, we consider the special case $d = k$. This means that at a corresponding node all the integer variables are fixed to certain values given by $r \in \mathbb{Z}^k$. In other words, a leaf node of the branch-and-bound tree is reached. At this point, the sets $\mathcal{L}$ of lower bounds and $\mathcal{U}$ of upper bounds for the enclosure of the nondominated set $\mathcal{N}$ of (MOMIQP) are built up as detailed in the following.

We initialize $\mathcal{L} = \emptyset, \mathcal{U} = U(\emptyset) = \{Z\}$, and $N = \emptyset$. At the leaf node corresponding to $r \in \mathbb{Z}^k$, we solve the problems $(P^r(\ell))$ for all $\ell \in L$. The optimal solutions $x^{*\ell,r} \in \mathbb{R}^{n-k}$ of (MOP^r) lead to feasible points $(r, x^{*\ell,r}) \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of (MOMIQP). The upper bound set $\mathcal{U}$ is then updated as

$$\mathcal{U} = U(N \cup \{f^r(x^{*\ell,r}) \mid \ell \in L\}). \tag{3}$$

More precisely, we use [21, Algorithm 3] (which is the same as [20, Algorithm 1]) to do so. By [20, Lemma 4.7], for the resulting local upper bound set it holds that

$$\mathcal{N} \subseteq \mathcal{U} - \mathbb{R}^m_+. \tag{4}$$

On the other hand, the lower bound set $\mathcal{L}$ is updated by

$$\mathcal{L} = \mathcal{L} \cup \{(\varphi^r(e^1), \ldots, \varphi^r(e^m))\}, \tag{5}$$

i.e., by the ideal point of the upper image set $\mathcal{P}^r$. We will show in Lemma 5.1 that the resulting set $\mathcal{L}$ computed by our algorithm is indeed a lower bound set in the sense of Definition 1. We remark that, while $\mathcal{U}$ is always an upper bound set for an enclosure in that sense, for $\mathcal{L}$ this only holds at the end of our algorithm.

4 Pruning of nodes

As already mentioned, given a certain node at level d of the branch-and-bound tree, the interval (I) defines the range of integer values for which corresponding child nodes at level $d+1$ need to be computed. In this section, we analyze how to stop the computation of new child nodes when fixing variable x_{d+1} to integer values outside this interval. In particular, we provide sufficient conditions that allow to consider a finite number of integer assignments. This implies that our method needs to explore only a finite number of nodes even in the case that the original problem has an unbounded feasible set.

4.1 Pruning by infeasibility

Whenever for some $d \in [k]$ and a vector $r = (r_1, \ldots, r_d) \in \mathbb{Z}^d$ the problem ($P^r(\ell)$) is infeasible, i.e., condition (Inf) holds, the corresponding node and all its children can of course be pruned:

Lemma 4.1 *Let the condition* (Inf) *hold for some $d \in [k]$ and some vector $r \in \mathbb{Z}^d$ of integer fixings. Then there is no feasible point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of* (MOMIQP) *such that $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_d)$.*

In Lemma 4.2 we prove that, in case (Inf) holds, for integer fixings $r \in \mathbb{Z}^d$ with $r_d > \lceil \beta(d) \rceil$ or $r_d < \lfloor \alpha(d) \rfloor$, thanks to linearity of the constraints, we can also prune the outer siblings of that node. Note that the condition (Inf) cannot occur for integer fixings $r \in \mathbb{Z}^d$ with $r_d \in [\, \alpha(d)\,,\, \beta(d)\,]$.

Lemma 4.2 *Let the condition* (Inf) *hold for some $d \in [k]$ and some vector $r \in \mathbb{Z}^d$ of integer fixings.*

(a) *If $r_d = \delta \geq \lceil \beta(d) \rceil$, then there is no feasible point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of* (MOMIQP) *such that $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_{d-1}, \bar{r}_d) =: \bar{r} \in \mathbb{Z}^d$ with $\bar{r}_d \geq \delta$.*
(b) *If $r_d = \delta \leq \lfloor \alpha(d) \rfloor$, then there is no feasible point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of* (MOMIQP) *such that $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_{d-1}, \bar{r}_d) =: \bar{r} \in \mathbb{Z}^d$ with $\bar{x}_d \leq \delta$.*

Proof We only prove (a). The proof of (b) is analogous.

Assume by contradiction that there exists a feasible point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of (MOMIQP) with $(\bar{x}_1, \ldots, \bar{x}_d) = \bar{r}$ and $\bar{r}_d \geq \delta$. By the definition of $\beta(d)$ there exists a feasible point of ($P^{r'}(\ell)$) for $r' := (r_1, \ldots, r_{d-1}) \in \mathbb{Z}^{d-1}$ and some $\ell \in L$. In particular, there exists some $x' \in \mathbb{R}^n$ with $Ax' \leq b$, $(x'_1, \ldots, x'_{d-1}) = (r_1, \ldots, r_{d-1})$, and $x'_d \leq \beta(d)$. For any $\lambda \in [0, 1]$ let $q(\lambda)$ be the point defined as

$$q(\lambda) := \lambda \bar{x} + (1 - \lambda) x'.$$

By linearity, it holds $Aq(\lambda) \leq b$ for all $\lambda \in [0, 1]$. Moreover, $q_i(\lambda) = r_i$ for all $i \in [d-1]$ and for all $\lambda \in [0, 1]$. For the d-th component of $q(\lambda)$ we have that $q_d(0) = x'_d \leq \beta(d) \leq \delta \leq \bar{x}_d = q_d(1)$. As a result, there exists $\bar{\lambda} \in [0, 1]$ such that $q_d(\bar{\lambda}) = \delta = r_d$. This contradicts (Inf). □

Of course, problem (MOMIQP) can still have an unbounded feasible set and the situation that (Inf) is satisfied might never occur. However, we will show in the forthcoming Sect. 4.2 that even in case (Inf) is never satisfied, we can still prune nodes and their siblings under certain conditions.

4.2 Pruning by lower and upper bounds

In this section, we analyze what happens when infeasibility does not occur. In particular, we make the following assumption.

Assumption 4.3 Let $d \in [k]$ and $r = (r_1, \ldots, r_d) \in \mathbb{Z}^d$ be a vector of integer fixings. Assume that (Inf) does not hold, i.e. $S^r = \{x \in \mathbb{R}^{n-d} \mid A^d x \leq b^r\} \neq \emptyset$.

In order to be able to prune certain nodes and their siblings as in Sect. 4.1, we define a pruning condition based on lower and upper bound sets. We say that $LB^r \subseteq \mathbb{R}^m$ is a lower bound set for the node corresponding to the vector $r \in \mathbb{Z}^d$ of integer fixings if

$$\{f(x) \in \mathbb{R}^m \mid x \in \mathbb{Z}^k \times \mathbb{R}^{n-k},\ (x_1, \ldots, x_d) = (r_1, \ldots, r_d),\ Ax \leq b\} \subseteq LB^r + \mathbb{R}^m_+.$$

A sufficient condition for this to hold is that $\mathcal{P}^r \subseteq LB^r + \mathbb{R}^m_+$. Due to the definition of $\varphi^r(\ell)$ and since $L \subseteq \mathbb{R}^m_+$, we have for any $\ell \in L$ that $\mathcal{P}^r \subseteq \{y \in \mathbb{R}^m \mid \ell^\top y \geq \varphi^r(\ell)\}$. Thus a valid lower bound set for the node is given by

$$LB^r := \left\{ y \in \mathbb{R}^m \;\middle|\; \ell^\top y \geq \varphi^r(\ell)\ \forall \ell \in L \right\}. \tag{6}$$

Further, due to $L \subseteq \mathbb{R}^m_+$, we have that $LB^r = LB^r + \mathbb{R}^m_+$. Since we assume that the set L contains the m unit vectors, we obtain for the ideal point $\mathrm{id}^r := (\varphi^r(e^1), \ldots, \varphi^r(e^m))$ of the set $\mathcal{P}^r$ that $LB^r \subseteq \{\mathrm{id}^r\} + \mathbb{R}^m_+$.

Intersecting the set of local upper bounds $\mathcal{U}$ with the lower bound set LB^r gives a pruning condition:

$$\forall\, u \in \mathcal{U} :\ u \notin LB^r. \tag{Cond}$$

We need the following lemma for proving that this is indeed a pruning condition.

Lemma 4.4 *Let Assumption 4.3 hold. If (Cond) holds then for the nondominated set $\mathcal{N}$ of (MOMIQP) we have $\mathcal{N} \cap LB^r = \emptyset$.*

Proof Since $L \subseteq \mathbb{R}^m_+$, for any $h \in -\mathbb{R}^m_+$ it holds that $\ell^\top h \leq 0$ for all $\ell \in L$. As a result, we have that $u \notin LB^r$ if and only if $(\{u\} - \mathbb{R}^m_+) \cap LB^r = \emptyset$. Hence, (Cond) holds if and only if $(\mathcal{U} - \mathbb{R}^m_+) \cap LB^r = \emptyset$. Together with (4) we obtain that if (Cond) holds then also $\mathcal{N} \cap LB^r = \emptyset$. $\square$

Since for any feasible point $\bar{x} \in \mathbb{Z}^d \times \mathbb{R}^{n-d}$ of (MOMIQP) with $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_d)$ it holds that $f(\bar{x}) \in LB^r$, we immediately conclude from Lemma 4.4 the following result for pruning:

Lemma 4.5 *Let Assumption 4.3 hold. Further, let $LB^r \in \mathbb{R}^m$ be a lower bound set as in (6) and let (Cond) hold. Then there is no efficient point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ for (MOMIQP) with $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_d)$.*

In the forthcoming Lemma 4.6, we prove that as soon as (Cond) holds for a node $r \in \mathbb{Z}^d$ with $r_d \notin [\lfloor \alpha(d) \rfloor, \lceil \beta(d) \rceil]$, we can prune its outer siblings $\bar{r} \in \mathbb{Z}^d$ with $(\bar{r}_1, \ldots, \bar{r}_{d-1}) = (r_1, \ldots, r_{d-1})$ and $\bar{r}_d > r_d$ or $\bar{r}_d < r_d$.

Lemma 4.6 *Let Assumption 4.3 hold for some $d \in [k]$ and some vector $r \in \mathbb{Z}^d$ of integer fixings with $r_d \notin [\lfloor \alpha(d) \rfloor, \lceil \beta(d) \rceil]$. Further, let $LB^r \in \mathbb{R}^m$ be the lower bound set computed as in (6) and let (Cond) hold.*

(a) *If $r_d = \delta > \lceil \beta(d) \rceil$, then there is no efficient point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of (MOMIQP) such that $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_{d-1}, \bar{r}_d) =: \bar{r} \in \mathbb{Z}^d$ with $\bar{r}_d \geq \delta$.*
(b) *If $r_d = \delta < \lfloor \alpha(d) \rfloor$, then there is no efficient point $\bar{x} \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ of (MOMIQP) such that $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_{d-1}, \bar{r}_d) =: \bar{r} \in \mathbb{Z}^d$ with $\bar{r}_d \leq \delta$.*

Proof We only prove (a). The proof of (b) is analogous.

If (Inf) holds for $\bar{r}$ then there cannot be an efficient point $\bar{x}$ of (MOMIQP) with $(\bar{x}_1, \ldots, \bar{x}_d) = \bar{r}$, see Lemma 4.1. Thus, in the following we consider the case where (Inf) does not hold, i.e., Assumption 4.3 holds for $\bar{r} \in \mathbb{Z}^d$. Then we can determine the set $LB^{\bar{r}}$ as in (6) based on the values $\varphi^{\bar{r}}(\ell)$ for $\ell \in L$. We will show that it holds

$$\varphi^r(\ell) \leq \varphi^{\bar{r}}(\ell) \quad \text{for all } \ \ell \in L \tag{7}$$

as then we have $LB^{\bar{r}} \subseteq LB^r$ and (Cond) also holds for $LB^{\bar{r}}$. Lemma 4.5 then concludes the proof.

To show (7), let $\ell \in L$ and denote by $\bar{f}^\ell : \mathbb{R}^{n-d+1} \to \mathbb{R}$ the function

$$\bar{f}^\ell(z) := \ell^\top f(r_1, \ldots, r_{d-1}, z_1, \ldots, z_{n-d+1}).$$

Within this proof, we use the notation $\bar{b} := b - A(r_1, \ldots, r_{d-1}, 0, \ldots, 0)$ and denote as usual by A^{d-1} the matrix which is obtained from A by deleting the first $d-1$ columns. Then $A(r_1, \ldots, r_{d-1}, z)^\top \leq b$ reduces to $A^{d-1}z \leq \bar{b}$. Using this notation, we have

$$\varphi^r(\ell) = \min_z \{\bar{f}^\ell(z) \mid z_1 = r_d, \ A^{d-1}z \leq \bar{b}, \ z \in \mathbb{R}^{n-d+1}\}$$

and

$$\varphi^{\bar{r}}(\ell) = \min_z \{\bar{f}^\ell(z) \mid z_1 = \bar{r}_d, \ A^{d-1}z \leq \bar{b}, \ z \in \mathbb{R}^{n-d+1}\}.$$

The first components of the unique minimal solutions $u^{*\ell} \in \operatorname{argmin}_z\{\bar{f}^\ell(z) \mid A^{d-1}z \leq \bar{b}, \ z \in \mathbb{R}^{n-d+1}\}$ determine the interval $[\lfloor \alpha(d) \rfloor, \lceil \beta(d) \rceil]$, cf. (2), and we have that

$$u_1^{*\ell} \leq \beta(d) \leq \lceil \beta(d) \rceil < \delta = r_d \leq \bar{r}_d.$$

For $\gamma \in \mathbb{R}$ with $u_1^{*\ell} < \gamma \leq \bar{r}_d$ we define the parametric optimization problem $P(\gamma)$ by

$$\begin{aligned} \min_z \quad & \bar{f}^\ell(z) \\ \text{s.t.} \quad & z_1 \geq \gamma, \\ & A^{d-1}z \leq \bar{b}, \\ & z \in \mathbb{R}^{n-d+1}. \end{aligned} \tag{$P(\gamma)$}$$

Since Assumption 4.3 holds for $\bar{r} \in \mathbb{Z}^d$, all of these optimization problems are solvable. We denote their optimal value by $v(\gamma)$ for $\gamma \in (u_1^{*\ell}, \bar{r}_d]$. Due to $\bar{r}_d \geq r_d$ we have that $v(r_d) \leq v(\bar{r}_d)$. Next, we prove by contradiction that for all $\gamma \in (u_1^{*\ell}, \bar{r}_d]$ it holds that

$$\begin{aligned} &\min_z \{\bar{f}^\ell(z) \mid z_1 \geq \gamma, \ A^{d-1}z \leq \bar{b}, \ z \in \mathbb{R}^{n-d+1}\} \\ &\quad = \min_z \{\bar{f}^\ell(z) \mid z_1 = \gamma, \ A^{d-1}z \leq \bar{b}, \ z \in \mathbb{R}^{n-d+1}\}. \end{aligned} \tag{8}$$

Let $\gamma \in (u_1^{*\ell}, \bar{r}_d]$ and $\bar{z} \in \mathbb{R}^{n-d+1}$ be the optimal solution of ($P(\gamma)$) with $\bar{z}_1 > \gamma$. We set $q(\lambda) := (1-\lambda)\bar{z} + \lambda u^{*\ell}$, i.e., $q_1(0) = \bar{z}_1 > \gamma$ and $q_1(1) = u_1^{*\ell} < \gamma$. Note that $A^{d-1}q(\lambda) \leq \bar{b}$

holds for all $\lambda \in [0, 1]$. Let $0 < \bar{\lambda} < 1$ be such that $q_1(\bar{\lambda}) = \gamma$. Moreover, by the definition of $u^{*\ell}$ we have $\bar{f}^\ell(u^{*\ell}) \le \bar{f}^\ell(\bar{z})$. Then, from the strict convexity of $\bar{f}^\ell$, we derive

$$\bar{f}^\ell(q(\bar{\lambda})) = \bar{f}^\ell((1-\bar{\lambda})\bar{z} + \bar{\lambda} u^{*\ell}) < (1-\bar{\lambda})\bar{f}^\ell(\bar{z}) + \bar{\lambda}\bar{f}^\ell(u^{*\ell}) \le \bar{f}^\ell(\bar{z}),$$

which contradicts the minimality of $\bar{z}$ for ($P(\gamma)$). Consequently, (8) holds, implies that $\varphi^r(\ell) = v(r_d) \le v(\bar{r}_d) = \varphi^{\bar{r}}(\ell)$, and we are done with showing (7). □

All pruning results within this subsection are based on the condition (Cond). The next result simplifies the evaluation of (Cond). It exploits the fact that for any $u \in \mathcal{U}$ it holds $u \notin LB^r$ if and only if there exists $\ell \in L$ with $\ell^\top u < \varphi^r(\ell)$.

Lemma 4.7 *Let Assumption 4.3 hold and define for $u \in \mathcal{U}$ the value $\sigma(u)$ by*

$$\sigma(u) := \min\{\ell^\top u - \varphi^r(\ell) \mid \ell \in L\}. \tag{9}$$

Then (Cond) holds if and only if $\sigma(u) < 0$ for all $u \in \mathcal{U}$.

The costs for evaluating (Cond) can be further reduced:

Remark 1 In order to verify whether (Cond) is satisfied it is sufficient to check whether $\sigma(u) < 0$ only for those $u \in \mathcal{U}$ with $u \ge \mathrm{id}^r$, where $\mathrm{id}^r \in \mathbb{R}^m$ is the ideal point of $\mathcal{P}^r$ or some underestimator of it. This holds because of $LB^r \subseteq \{\mathrm{id}^r\} + \mathbb{R}^m_+$. The ideal point is obtained as a byproduct when calculating LB^r.

As we are making use of dual formulations of the problems ($P^r(\ell)$) (see Sect. 6) and as we will try to avoid to solve them exactly, we sometimes calculate just lower bounds for $\varphi^r(\ell)$ in (6) and thus derive only sets LB' which are supersets of LB^r. Still those sets can be used to formulate a sufficient condition for (Cond):

Remark 2 Let $LB^r \in \mathbb{R}^m$ be the lower bound set computed as in (6) and let $LB' \supseteq LB^r$ be an arbitrary superset of it. Then, if (Cond) holds for LB', i.e. if for all $u \in \mathcal{U}$ we have $u \notin LB'$, then (Cond) holds also for LB^r.

In Sect. 6, we explain how to make use of Lemma 4.7 in combination with Remark 2 to speed up the pruning strategy in our branch-and-bound algorithm.

4.3 Occurring of pruning conditions

In the last two subsections, we formulated conditions for pruning a node in case (Inf) or (Cond) hold. Furthermore, we have given conditions in order to prune all the outer siblings of a node in case (Inf) or (Cond) are satisfied at that node. However, since we are not assuming boundedness of the feasible region of (MOMIQP), it may happen that an infinite number of nodes is visited, as neither (Inf) nor (Cond) are satisfied at any node. This would imply that our algorithm never stops. The following lemma shows that this cannot happen and that for all $d \in [k]$ there exist only finitely many integer fixings $r \in \mathbb{Z}^d$ such that neither (Inf) nor (Cond) are satisfied. The strong convexity of the objective functions is key to the proof of the result.

Lemma 4.8 *Let $d \in [k-1] \cup \{0\}$ and let Assumption 4.3 hold at level d for $r \in \mathbb{Z}^d$, i.e., (Inf) does not hold. Then there exists $\gamma \in \mathbb{Z}$ such that for all $\bar{r} \in \mathbb{Z}^{d+1}$ with $(\bar{r}_1, \dots, \bar{r}_d) = r$ and*

$$\bar{r}_{d+1} \notin \left[\lfloor \alpha(d+1) \rfloor - \gamma, \lceil \beta(d+1) \rceil + \gamma\right]$$

either (Cond) or (Inf) is satisfied.

Proof First, select an arbitrary element $\ell \in L$. For the current finite and nonempty set of local upper bounds $\mathcal{U}$ define $\xi := \max\{\ell^\top u \mid u \in \mathcal{U}\}$. Since the objective functions f_j, $j \in [m]$, are strongly convex, $\ell^\top f : \mathbb{R}^n \to \mathbb{R}$ is a strongly convex quadratic function, too. Thus, there exists some $\nu > 0$ such that

$$\nu\lambda(1-\lambda)\left\|x - x'\right\|_2^2 + \ell^\top f(\lambda x + (1-\lambda)x') \leq \lambda\ell^\top f(x) + (1-\lambda)\ell^\top f(x') \tag{10}$$

for all $x, x' \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$. Let $x^{*\ell,r}$ denote the unique minimizer of the problem ($P^r(\ell)$). By definition it holds $\alpha(d+1) \leq x_1^{*\ell,r}$ and $\beta(d+1) \geq x_1^{*\ell,r}$.

Now, let $\delta \geq 0$ and consider the optimization problem

$$\begin{aligned} \min_x \quad & \ell^\top f^r(x) \\ \text{s.t.} \quad & A^d x \leq b^r \\ & x_1 = \lceil \beta(d+1) \rceil + \delta \\ & x \in \mathbb{R}^{n-d}. \end{aligned} \tag{11}$$

First, assume that there exists some $\bar{\delta} \geq 0$ such that (11) is infeasible. Then we define $\bar{\gamma} := \lceil \bar{\delta} \rceil \in \mathbb{N}$ and it holds that (11) remains infeasible for all $\delta \geq \bar{\gamma} \geq \bar{\delta}$. To see this, assume that there exists some $\delta' > \bar{\delta}$ such that (11) is feasible. The corresponding minimizer $x' \in \mathbb{R}^{n-d}$ is not only feasible for (11), but also for ($P^r(\ell)$). Further, it holds that

$$x_1^{*\ell,r} \leq \lceil \beta(d+1) \rceil + \bar{\delta} < \lceil \beta(d+1) \rceil + \delta' = x_1'.$$

However, since both $x^{*\ell,r}$ and x' are feasible for ($P^r(\ell)$) and the constraints are all linear, there exists some feasible point $\hat{x} \in \mathbb{R}^{n-d}$ for ($P^r(\ell)$) with $\hat{x}_1 = \lceil \beta(d+1) \rceil + \bar{\delta}$ which contradicts the assumption that (11) is infeasible for $\bar{\delta}$.

Next, assume that (11) is feasible for all $\delta \geq 0$ and denote for each $\delta \geq 0$ by $\bar{x}(\delta) \in \mathbb{R}^{n-d}$ its unique minimizer. Note that for $\bar{r} \in \mathbb{Z}^{d+1}$ with $(\bar{r}_1, \ldots, \bar{r}_d) = (r_1, \ldots, r_d)$ and $\bar{r}_{d+1} = \lceil \beta(d+1) \rceil + \delta$ it holds $\varphi^{\bar{r}}(\ell) = \ell^\top f^r(\bar{x}(\delta))$.

We obtain from (10) with $\lambda = 1/2$ and for any $\delta \geq 0$ that

$$0.25\,\nu \left\|x^{*\ell,r} - \bar{x}(\delta)\right\|_2^2 + \ell^\top f^r(0.5\,x^{*\ell,r} + 0.5\,\bar{x}(\delta)) \leq 0.5\,\ell^\top f^r(x^{*\ell,r}) + 0.5\,\ell^\top f^r(\bar{x}(\delta)).$$

Since $\bar{x}(\delta)$ and $x^{*\ell,r}$ are feasible for ($P^r(\ell)$), so are all convex combinations of them and in particular the point $0.5\,x^{*\ell,r} + 0.5\,\bar{x}(\delta)$. As $x^{*\ell,r}$ is the unique minimizer of ($P^r(\ell)$) we have $\ell^\top f^r(x^{*\ell,r}) \leq \ell^\top f^r(0.5\,x^{*\ell,r} + 0.5\,\bar{x}(\delta))$. Further, we have $\ell^\top f^r(x^{*\ell,r}) \leq \ell^\top f^r(\bar{x}(\delta))$ and hence

$$0.25\,\nu \left\|x^{*\ell,r} - \bar{x}(\delta)\right\|_2^2 + \ell^\top f^r(x^{*\ell,r}) \leq \ell^\top f^r(\bar{x}(\delta)).$$

Finally, making use of $\bar{x}(\delta)_1 = \lceil \beta(d+1) \rceil + \delta \geq x_1^{*\ell,r} + \delta$, we obtain that

$$0.25\nu\delta^2 + \ell^\top f^r(x^{*\ell,r}) \leq \ell^\top f^r(\bar{x}(\delta)).$$

For $\delta \geq 0$ larger than some $\bar{\gamma} \in \mathbb{N}$, we have that the left hand side of this inequality exceeds ξ such that $\xi < \ell^\top f^r(\bar{x}(\delta))$ for all $\delta \geq \bar{\gamma}$.

Analogously, replacing the constraint $x_1 = \lceil \beta(d+1) \rceil + \delta$ in (11) by $x_1 = \lfloor \alpha(d+1) \rfloor - \delta$ and using that $\alpha(d+1) \leq x_1^{*\ell,r}$, one obtains that there exists some $\underline{\gamma} \in \mathbb{N}$ such that either (11) becomes infeasible or $\xi < \ell^\top f^r(\bar{x}(\delta))$ for all $\delta \geq \underline{\gamma}$.

Thus, for $\gamma := \max\{\bar{\gamma}, \underline{\gamma}\}$ and an arbitrary vector $\bar{r} \in \mathbb{Z}^{d+1}$ of integer fixings with $(\bar{r}_1, \ldots, \bar{r}_d) = (r_1, \ldots, r_d)$ and $\bar{r}_{d+1} \notin \left[\lfloor \alpha(d+1) \rfloor - \gamma, \lceil \beta(d+1) \rceil + \gamma\right]$ we obtain that either (Cond) holds since $\varphi^{\bar{r}}(\ell) > \xi \geq \ell^\top u$ for all $u \in \mathcal{U}$ or that (Inf) holds. $\square$

5 `DEIA-BB`: algorithmic scheme and finiteness

In order to summarize what we have presented so far, we report in Algorithm 2 the scheme of our branch-and-bound method, called `DEIA-BB` (for *Detector of Efficient Integer Assignments-Branch-and-Bound*). As already mentioned, `DEIA-BB` computes two things. Primarily, it computes a lower bound set $\mathcal{L}$ and an upper bound set $\mathcal{U}$ for an enclosure of the nondominated set of (MOMIQP). Thereby, it also computes a superset $\mathcal{S}$ of the set of efficient integer assignments. In the following, we will briefly describe step-by-step how the algorithm works and how the steps are related to the theoretical results presented in the previous sections.

Algorithm 2 `DEIA-BB`: Detector of Efficient Integer Assignments

INPUT: m strongly convex quadratic functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, m$, linear constraints $Ax \leq b$, finite set $L \subseteq \mathbb{R}^m_+$ with $e^j \in L$ for all $j \in [m]$
OUTPUT: $\mathcal{L}, \mathcal{U}, \mathcal{S}$

1: Perform a preprocessing phase to speed up computations (see Algorithm 3)
2: Set $U = U(\emptyset) = \{Z\}$, $\mathcal{S} = \emptyset$ and $d = 0$
3: **if** $\{x \in \mathbb{R}^n \mid Ax \leq b\} \neq \emptyset$ **then**
4: Compute $\alpha(1)$ and $\beta(1)$ according to (2)
5: Set $d = 1,\ r_1 = \lfloor \alpha(1) \rfloor$
6: **else**
7: Stop the algorithm and state infeasibility of (MOMIQP)
8: **end if**
9: **while** $d > 0$ **do**
10: Evaluate whether (Inf) or (Cond) holds for d and $r = (r_1, \dots, r_d)$
11: **if** not ((Cond) or (Inf)) **then**
12: **if** $d = k$ **then**
13: Update $\mathcal{U}$ and $\mathcal{L}$ according to (3), (5)
14: Update $\mathcal{S} = \mathcal{S} \cup \{r\}$
15: **else**
16: Compute $\alpha(d+1)$ and $\beta(d+1)$ according to (2)
17: **end if**
18: **end if**
19: **if** ((Cond) or (Inf)) and $r_d < \lfloor \alpha(d) \rfloor$ **then**
20: Set $d = d - 1$;
21: **if** $(d > 0)$ Update r_d with Algorithm 1 **end if**
22: **else if** ((Cond) or (Inf)) and $r_d \geq \lfloor \alpha(d) \rfloor$ **then**
23: **if** $r_d \geq \lceil \beta(d) \rceil$ **then**
24: Set $r_d = \lfloor \alpha(d) \rfloor - 1$
25: **else**
26: Set $r_d = r_d + 1$
27: **end if**
28: **else if** $d \leq k - 1$ **then**
29: Set $d = d + 1$;
30: Set $r_d = \lfloor \alpha(d) \rfloor$
31: **else**
32: Update r_d with Algorithm 1
33: **end if**
34: **end while**

The algorithm first checks in line 3 whether the continuous relaxation of (MOMIQP) is feasible, i.e., whether the corresponding feasible set $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ is nonempty. If it is nonempty, then $\alpha(1)$ and $\beta(1)$ are computed and the algorithm continues with the main loop in line 9. Otherwise, the algorithm detects the infeasibility of (MOMIQP) and stops.

The while loop basically computes feasible leaf nodes, i.e., integer fixings $\bar{r} \in \mathbb{Z}^k$ such that $\{x \in \mathbb{R}^{n-k} \mid A^k x \leq b^{\bar{r}}\} \neq \emptyset$, with a depth-first approach. It starts at depth $d = 1$ with the integer fixing $r = r_1 = \lfloor \alpha(1) \rfloor \in \mathbb{Z}^d$. We remark that the depth $d \in \mathbb{N}$ never exceeds k since it is only increased in line 29 of the algorithm and this line is only called if $d \leq k - 1$. We also remark that the first node that is considered at level $d + 1$ is always the one with the vector $r = (r_1, \dots, r_d, \lfloor \alpha(d + 1) \rfloor)$ of integer fixings.

For an arbitrary node, i.e., an arbitrary vector $r \in \mathbb{Z}^d$ of integer fixings, at depth $d \in [k]$ the algorithm checks (line 11) whether this node needs to be explored. Namely, it checks whether (MOMIQP) with the first d variables fixed to $r \in \mathbb{Z}^d$ is feasible and (Cond) does not hold. If this is the case and $d = k$ the algorithm has reached a leaf node and thus a feasible integer fixing for (MOMIQP). This allows us to update the lower and upper bound sets $\mathcal{L}$ and $\mathcal{U}$ for the initial enclosure, see line 13. Otherwise, we have not reached a leaf node and compute the bounds $\alpha(d + 1)$ and $\beta(d + 1)$ for the integer fixings at level $d + 1$.

Next, the algorithm checks whether the children or siblings of the current node can be pruned based on the results of Sects. 4.1 and 4.2. Note that for the former we need (Inf) and for the latter we need (Cond) to hold in order to prune. If neither (Inf) nor (Cond) hold, then the conditions in lines 19 and 22 of Algorithm 2 are not satisfied, nothing is pruned, and the algorithm moves on (with its depth-first approach) to level $d + 1$, see line 29. In case $d = k$, i.e., at a leaf node, the algorithm stays at level $d = k$ and explores the siblings of the current node, see line 32. We remark that by Lemma 4.8 at each level $d \in [k]$ there only exist finitely many nodes where neither (Inf) nor (Cond) are satisfied.

Thus, for Algorithm 2, it remains the setting that (Inf) or (Cond) are satisfied, i.e., that either the clause in line 19 or in line 22 is true. If (Inf) holds then we can prune all the children nodes of the current node corresponding to $r \in \mathbb{Z}^d$ and in particular the node itself. Also if (Cond) holds we can prune all the children nodes by Lemma 4.5. Thus the algorithm only needs to decide whether siblings of the current node need to be explored or can be pruned. Since at each level d we always start with $r_d = \lfloor \alpha(d) \rfloor$ and the value of r_d is only changed if one of the clauses in line 19 or 22 is true, we first consider the case in line 22. If $\lfloor \alpha(d) \rfloor \leq r_d \leq \lceil \beta(d) \rceil$ we cannot prune any siblings of the current node based on the results from the previous sections. Thus, we need to explore them. This is done by setting $r_d = r_d + 1$ in line 26 of Algorithm 2. If $r_d > \lceil \beta(d) \rceil$ it is known from Lemmas 4.2 and 4.6 that all siblings with $\tilde{r}_d > r_d$ can be pruned. Thus, the algorithm will not continue to explore such nodes and goes on exploring nodes to the left of $\lfloor \alpha(d) \rfloor$ by setting $r_d = \lfloor \alpha(d) \rfloor - 1$, see line 24. For any siblings of the current node this means that the condition in line 22 will never be satisfied again. Instead, the condition in line 19 will be satisfied for any future sibling where (Inf) or (Cond) holds. If this is the case, then also all siblings with $\tilde{r}_d < r_d$ can be pruned by Lemma 4.2 or 4.6. As a result, since the node corresponding to the current integer assignment $r \in \mathbb{Z}^d$ itself and all of its sibling nodes can be pruned, also its parent node at level $d - 1$ can be pruned. Thus, Algorithm 2 moves back to level $d - 1$, see line 20, and continues by exploring a sibling of that parent node.

This whole procedure is repeated until we reach line 20 with $d = d - 1 = 0$, return to the root node, and terminate the algorithm since the condition $d > 0$ of the while loop is no longer satisfied. Together with Lemma 4.8 and the fact that each of the intervals $[\lfloor \alpha(d + 1) \rfloor, \lceil \beta(d + 1) \rceil]$ is bounded, we eventually reach this line within a finite number of iterations and obtain the following finiteness result for Algorithm 2.

Theorem 1 *Algorithm 2 stops after a finite number of iterations returning the sets $\mathcal{L}$, $\mathcal{U}$ and $\mathcal{S}$.*

From the sets $\mathcal{L}$ and $\mathcal{U}$ we can build an enclosure of the nondominated set $\mathcal{N}$ of (MOMIQP). Indeed, the following lemma shows that the output set $\mathcal{L} \subseteq \mathbb{R}^m$ of `DEIA-BB` is a lower bound set of (MOMIQP). This in turn implies (see Corollary 5.2) that `DEIA-BB` is able to release an enclosure (or box approximation) in a finite number of iterations.

Lemma 5.1 *The set $\mathcal{L} \subseteq \mathbb{R}^m$ computed by Algorithm 2 is a lower bound set in the sense of Definition 1.*

Proof By Theorem 1 we know that $\mathcal{L}$ is finite. It only remains to prove that $\mathcal{N} \subseteq \mathcal{L} + \mathbb{R}^m_+$. So let $z \in \mathcal{N}$ and let $\bar{x} \in S,\ r \in \mathbb{Z}^k$, with $(\bar{x}_1, \dots, \bar{x}_k) = (r_1, \dots, r_k)$, such that $f(\bar{x}) = z$. We prove that `DEIA-BB` has explored the leaf node $r \in \mathbb{Z}^k$. This would imply by (5) that for the ideal point $\mathrm{id}^r = (\varphi^r(e^1), \dots, \varphi^r(e^m)) \in \mathcal{L}$ computed at this leaf node it holds that $\mathrm{id}^r \leq z$. By contradiction, assume that the leaf node $r \in \mathbb{Z}^k$ has not been explored. Then the parent node $r' \in \mathbb{Z}^d$ of this leaf node at a certain level $d \in [k]$ with $(r'_1, \dots, r'_d) = (r_1, \dots, r_d)$ was pruned. Since $\bar{x} \in \mathbb{R}^n$ is feasible for (MOMIQP) this parent node was not pruned by (Inf). Hence, it was pruned by (Cond). This means that for all $u \in \mathcal{U}$ we have that $u \notin LB^{r'}$. By (4) we also have that $z = f(\bar{x}) \in \mathcal{N} \subseteq \mathcal{U} - \mathbb{R}^m_+$. But then there exists $u \in \mathcal{U}$ such that $u \geq z \in LB^r + \mathbb{R}^m_+ = LB^r \subseteq LB^{r'}$ which is a contradiction. □

Corollary 5.2 *Let $\mathcal{L}, \mathcal{U} \subseteq \mathbb{R}^m$ be the finite sets obtained by Algorithm 2, $\sigma > 0$ a small offset, and denote by $e \in \mathbb{R}^m$ the all-ones vector. Then $B := \mathcal{A}(L', U') = (L' + \mathbb{R}^m_+) \cap (U' - \mathbb{R}^m_+)$ with $L' := \mathcal{L} - \{\sigma e\}$ and $U' := \mathcal{U} + \{\sigma e\}$ is an enclosure of the nondominated set $\mathcal{N}$ of* (MOMIQP) *such that $\mathcal{N} \subseteq \mathrm{int}(B)$.*

6 Using dual relaxations

Let $r = (r_1, \dots, r_d)^\top \in \mathbb{Z}^d$ be the vector of integer fixings defining a node at level d in our branch-and-bound algorithm. As already stated, the multiobjective continuous relaxation of problem (MOMIQP), where the integer variables are fixed to $r \in \mathbb{R}^d$, is (MOP^r). Instead of addressing the problem (MOP^r), our aim is to compute LB^r by addressing the dual of the $|L|$ single-objective problems ($P^r(\ell)$), where the objective function is defined by

$$\begin{aligned}\ell^\top f^r(x) &= \sum_{j=1}^m \ell_j (x^\top Q_j^d x + (c^{j,r})^\top x + a_{j,r}) \\ &= x^\top \left(\sum_{j=1}^m \ell_j Q_j^d \right) x + \left(\sum_{j=1}^m \ell_j c^{j,r} \right)^\top x + \sum_{j=1}^m \ell_j a_{j,r}.\end{aligned}$$

We do so in order to accelerate the pruning process for those nodes that can be pruned because of (Cond). As we will see, addressing the dual of the $|L|$ single-objective problems ($P^r(\ell)$) may allow to stop the computation of LB^r to a rough though effective-for-pruning set and this clearly helps in saving computational effort.

For ease of notation, we introduce the following:

$$\bar{Q}_\ell^d = \sum_{j=1}^m \ell_j Q_j^d, \qquad \bar{c}^{\ell,r} = \sum_{j=1}^m \ell_j c^{j,r}, \qquad \bar{a}_{\ell,r} = \sum_{j=1}^m \ell_j a_{j,r}.$$

We obtain the dual of problem ($P^r(\ell)$) by first forming the Lagrangian $\mathcal{L}_\ell^d \colon \mathbb{R}^{n-d} \times \mathbb{R}^p \to \mathbb{R}$,

$$\mathcal{L}_\ell^d(x, \lambda) = x^\top \bar{Q}_\ell^d x + (\bar{c}^{\ell,r})^\top x + \bar{a}_{\ell,r} + \lambda^\top (A^d x - b^r)$$

which depends on x and the Lagrange multipliers $\lambda \in \mathbb{R}^p$. Then, for fixed $\lambda \in \mathbb{R}^p$, one has to minimize $\mathcal{L}_\ell^d$ with respect to the primal variables x. As $\bar{Q}_\ell^d$ is under our assumptions positive definite, $\mathcal{L}_\ell^d(\cdot, \lambda)$ is a strictly convex quadratic function and its unique minimizer can be computed in closed form as

$$x_\ell^d(\lambda) = -\frac{1}{2}(\bar{Q}_\ell^d)^{-1}(\bar{c}^{\ell,r} + A^{d\top}\lambda).$$

Then, the dual of problem ($P^r(\ell)$) is

$$\begin{aligned} \max_{\lambda} \quad & \mathcal{L}_\ell^d(\lambda) \\ & \lambda \in \mathbb{R}_+^p, \end{aligned} \tag{12}$$

with

$$\begin{aligned} \mathcal{L}_\ell^d(\lambda) &:= \mathcal{L}_\ell^d(x_\ell^d(\lambda), \lambda) \\ &= \lambda^\top\left(-\frac{1}{4}A^d(\bar{Q}_\ell^d)^{-1}A^{d\top}\right)\lambda - \left(b^{r\top} + \frac{1}{2}(\bar{c}^{\ell,r})^\top(\bar{Q}_\ell^d)^{-1}A^{d\top}\right)\lambda \\ &\quad - \frac{1}{4}(\bar{c}^{\ell,r})^\top(\bar{Q}_\ell^d)^{-1}(\bar{c}^{\ell,r}) + \bar{a}_{\ell,r}. \end{aligned}$$

Note that $-\frac{1}{4}A^d(\bar{Q}_\ell^d)^{-1}A^{d\top} =: -\widetilde{Q}_\ell^d$ is a negative semidefinite matrix, so that problem (12) can be seen as a convex quadratic minimization problem with simple nonnegativity constraints. Also note that since all constraints of problem ($P^r(\ell)$) are affine, strong duality holds if the primal problem ($P^r(\ell)$) is feasible [26]. On the other hand, if problem ($P^r(\ell)$) is infeasible, the dual problem (12) is unbounded [26].

Thanks to weak duality, we also have that $\mathcal{L}_\ell^d(\lambda) \leq \varphi^r(\ell)$ for each feasible $\lambda \geq 0$. This means that Lemma 4.7 can be easily extended, cf. Remark 2, as follows:

Lemma 6.1 *Let Assumption 4.3 hold and define for $u \in \mathcal{U}$, $\lambda \in \mathbb{R}_+^p$ the value $\sigma_{\mathcal{L}}(u, \lambda)$ by*

$$\sigma_{\mathcal{L}}(u, \lambda) := \min\{\ell^\top u - \mathcal{L}_\ell^d(\lambda) \mid \ell \in L\}.$$

Then (Cond) holds if and only if for all $u \in \mathcal{U}$ there exists some $\lambda := \lambda(u) \in \mathbb{R}_+^p$ such that $\sigma_{\mathcal{L}}(u, \lambda) < 0$.

Proof First we will show that if for all $u \in \mathcal{U}$ there exists $\lambda(u) \in \mathbb{R}_+^p$ such that $\sigma_{\mathcal{L}}(u, \lambda(u)) < 0$ then this implies (Cond). Assume by contradiction that (Cond) does not hold. Then there exists $\bar{u} \in \mathcal{U}$ such that $\bar{u} \in LB^r$. This implies $\ell^\top\bar{u} - \varphi^r(\ell) \geq 0$ for all $\ell \in L$. Due to weak duality it holds for all $\lambda \in \mathbb{R}_+^p$ that $\mathcal{L}_\ell^d(\lambda) \leq \varphi^r(\ell)$. Hence, we also have that $\ell^\top\bar{u} - \mathcal{L}_\ell^d(\lambda) \geq 0$ for all $\ell \in L$ and all $\lambda \in \mathbb{R}_+^p$ which contradicts $\sigma_{\mathcal{L}}(\bar{u}, \lambda(\bar{u})) < 0$.

We now show that if (Cond) holds, then for all $u \in \mathcal{U}$ there exists $\lambda \in \mathbb{R}_+^p$ such that $\sigma_{\mathcal{L}}(u, \lambda) < 0$. For that fix some $u \in \mathcal{U}$. By Lemma 4.7, there exists some $\ell \in L$ such that $\ell^\top u < \varphi^r(\ell)$. Since strong duality holds, $\hat{\lambda}^\ell \in \mathbb{R}_+^p$ exists such that $\mathcal{L}_\ell^d(\hat{\lambda}^\ell) = \varphi^r(\ell)$. This implies that also $\sigma_{\mathcal{L}}(u, \hat{\lambda}^\ell) < 0$. □

Remark 3 If Assumption 4.3 does not hold, namely ($P^r(\ell)$) is infeasible, for each $\ell \in L$ a sequence of points $\{\lambda^{\ell,k}\} \subseteq \mathbb{R}_+^p$ exists such that $\lim_{k\to\infty}\mathcal{L}_\ell^d(\lambda^{\ell,k}) = +\infty$. In particular, for each $\ell \in L$ there is some sufficiently large $\bar{k}(\ell) \in \mathbb{N}$ such that

$$\left(\max_{u\in\mathcal{U}} \ell^\top u\right) - \mathcal{L}_\ell^d(\lambda^{\ell,\bar{k}(\ell)}) < 0.$$

Thus, for all $u \in \mathcal{U}$ and all $\ell \in L$ there is some $\lambda := \lambda^{\ell,\bar{k}(\ell)}$ such that $\ell^\top u - \mathcal{L}_\ell^d(\lambda) < 0$ which implies that for each $u \in \mathcal{U}$ there is some $\lambda \in \mathbb{R}^p_+$ with $\sigma_{\mathcal{L}}(u, \lambda) < 0$. In fact, there even exists one $\lambda' \in \mathbb{R}^p_+$ for all $u \in \mathcal{U}$ such that $\sigma_{\mathcal{L}}(u, \lambda') < 0$.

We address problem (12) with `FAST-QPA`, an active set feasible method devised in [25] that uses conjugate gradient directions. The reduced matrices $\bar{Q}_\ell^d$, $(\bar{Q}_\ell^d)^{-1}$, and A^d only depend on the depth d, but not on specific integer fixings $r \in \mathbb{Z}^d$. Hence, the quadratic part of the reduced dual objective functions $\widetilde{Q}_\ell^d$ can be computed in the preprocessing phase, as it only depends on $(\bar{Q}_\ell^d)^{-1}$ and A^d. What is more, also the maximum eigenvalue $\lambda_{\max}(\widetilde{Q}_\ell^d)$, needed for ensuring a proper setting of the parameter for the active set estimate and the convergence of `FAST-QPA` (see [25]), can be computed in the preprocessing phase. The preprocessing phase used in our implementation is detailed in Algorithm 3.

Algorithm 3 Preprocessing

INPUT: m strongly convex quadratic functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, m$, linear constraints $Ax \leq b$, finite set of vectors L, number of integer variables k
OUTPUT: $(\bar{Q}_\ell^d)$, $(\bar{Q}_\ell^d)^{-1}$, A^d, $\widetilde{Q}_\ell^d$, $\lambda_{max}(\widetilde{Q}_\ell^d)$ for $d = 0, \dots, n-1$, for $\ell \in L$;

1: For $d = 0, \dots, n-1$ let A^d be the submatrix of A given by columns $d+1, \dots, n$;
2: For $d = 0, \dots, n-1$ and $\ell \in L$ compute the submatrix $\bar{Q}_\ell^d$;
3: For $d = 0, \dots, n-1$ and $\ell \in L$ compute $(\bar{Q}_\ell^d)^{-1}$;
4: For $d = 0, \dots, n-1$ and $\ell \in L$ compute $\widetilde{Q}_\ell^d = A^d(\bar{Q}_\ell^d)^{-1}A^{d\top}$
5: For $d = 0, \dots, n-1$ and $\ell \in L$ compute $\lambda_{\max}(\widetilde{Q}_\ell^d)$, the maximum eigenvalue of $\widetilde{Q}_\ell^d$,

Let $\{\lambda^k\}$ be the sequence of points produced by `FAST-QPA` when dealing with problem (12). Given the properties of `FAST-QPA`, λ^k is feasible for all $k \in \mathbb{N}$ and $\{\mathcal{L}_\ell^d(\lambda^k)\}$ is a monotonically increasing sequence. From the convergence results shown in [25, Proposition 11], in case problem (12) admits a maximal solution, under specific assumptions on the parameter used in the active set estimate, we have that

$$\lim_{k \to +\infty} \| \max\{0, \nabla \mathcal{L}_\ell^d(\lambda^k)\}\| = 0.$$

By [25, Theorem 13] this implies that every limit point of the sequence $\{\lambda^k\}$ produced by `FAST-QPA` satisfies the standard first-order optimality conditions for problem (12). Furthermore, since problem (12) is a convex problem (maximization of a concave function over a convex feasible set), this in turn implies that every limit point of $\{\lambda^k\}$ is an optimal point. In our implementation of `FAST-QPA`, we declare optimality when the point λ^k satisfies the condition

$$\| \max\{0, \nabla \mathcal{L}_\ell^d(\lambda^k)\}\| \leq 10^{-5}, \tag{13}$$

having then a guarantee that the algorithm stops after a finite number of iterations.

Handling problem (12) with a feasible method (i.e., an optimization method able to produce a sequence of feasible points) allows us to implement a strategy for which the node corresponding to $r \in \mathbb{Z}^d$ can be pruned before computing the lower bound set LB^r. We call this phenomenon *early pruning*. We now describe the implemented strategy and give an example of early pruning.

Given $u \in \mathcal{U}$, thanks to Lemma 6.1 and Remark 3, when dealing with problem (12) for a specific $\ell \in L$, we stop `FAST-QPA` as soon as one of the following occurs:

i) we get, at iteration $\hat{k}(\ell)$, that $\ell^\top u < \mathcal{L}_\ell^d(\lambda^{\ell,\hat{k}(\ell)})$ implying

$$\sigma_{\mathcal{L}}(u, \lambda^{\ell,\hat{k}(\ell)}) < 0, \tag{14}$$

ii) we have that (13) holds at iteration $k(\ell)$ and we set $\varphi^r(\ell) := \mathcal{L}_\ell^d(\lambda^{\ell,k(\ell)})$.

Note that, in case i), $\hat{k}(\ell) \le k(\ell)$ and $\mathcal{L}_\ell^d(\lambda^{\ell,\hat{k}(\ell)}) \le \mathcal{L}_\ell^d(\lambda^{\ell,k(\ell)})$. If Assumption 4.3 holds for the current vector $r \in \mathbb{Z}^d$ of integer fixings at level d, i.e., $S^r \neq \emptyset$, and (14) holds for every $u \in \mathcal{U}$, condition (Cond) holds and the node can be pruned. Moreover, in case (Inf) holds, i.e., $S^r = \emptyset$, the $|L|$ dual problems (12) are unbounded. Then, i) occurs and (14) is satisfied for every $u \in \mathcal{U}$ so that the node is pruned after a finite number of iterations of `FAST-QPA` in that case as well. Consequently, by applying `FAST-QPA` in our implementation of `DEIA-BB`, we can possibly prune the node using only $\hat{k}(\ell)$ iterations of `FAST-QPA`, see i), instead of $k(\ell)$ iterations to compute $\varphi^r(\ell)$ exactly, see ii).

We now discuss a biobjective example where *early pruning* is possible, see also Figure 1. Consider an integer fixing $r = (r_1, \ldots, r_d) \in \mathbb{Z}^d$, a set of local upper bounds $\mathcal{U} = \{u^1, u^2, u^3\}$, and a set of vectors $L = \{(1,0),\ (0,1),\ (0.5, 0.5)\}$. In Figure 1a the optimal lower bound set LB^r is represented by the blue dashed lines. We will show by this example that our pruning strategy allows us to prune the node without the need of computing LB^r exactly, but just a rough approximation of it. At the beginning, `DEIA-BB` takes into account the first local upper bound $u^1 \in \mathcal{U}$ and calls `FAST-QPA` on problem (12) with $\ell^1 = (1,0)^\top$. In Figure 1b we see how `FAST-QPA` stops when i) is satisfied. In other words, for u^1, `FAST-QPA` stops at iteration $\hat{k}(\ell^1)$ and detects a $\lambda^{\ell^1,\hat{k}(\ell^1)}$ such that $\sigma_{\mathcal{L}}(u^1, \lambda^{\ell^1,\hat{k}(\ell^1)}) < 0$. Since $\sigma_{\mathcal{L}}(u^1, \lambda^{\ell^1,\hat{k}(\ell^1)}) < 0$, `DEIA-BB` can move to the next local upper bound $u^2 \in \mathcal{U}$. Since $\sigma_{\mathcal{L}}(u^2, \lambda^{\ell^1,\hat{k}(\ell^1)}) > 0$, `FAST-QPA` is resumed on problem (12) with $\ell^1 = (1,0)^\top$, from iteration $\hat{k}(\ell^1)$. From Figure 1c we see again how `FAST-QPA` stops before reaching optimality or, in other words, it stops at an iteration $\bar{k}(\ell^1) > \hat{k}(\ell^1)$ such that $\sigma_{\mathcal{L}}(u^2, \lambda^{\ell^1,\bar{k}(\ell^1)}) < 0$. As before, since $\sigma_{\mathcal{L}}(u^2, \lambda^{\ell^1,\bar{k}(\ell^1)}) < 0$, `DEIA-BB` can move to the next local upper bound $u^3 \in \mathcal{U}$.

Since $\sigma_{\mathcal{L}}(u^3, \lambda^{\ell^1,\bar{k}(\ell^1)}) > 0$, `FAST-QPA` is resumed on problem (12) with $\ell^1 = (1,0)^\top$, from iteration $\bar{k}(\ell^1)$. From Figure 1d we notice (see the blue dashed line) that in this case `FAST-QPA` stops reaching optimality, or in other words at an iteration $k(\ell^1)$ such that (13) holds. However, we still have that $\sigma_{\mathcal{L}}(u^3, \lambda^{\ell_1,k(\ell^1)}) > 0$. Then, the new vector $\ell^2 \in L$ is considered and `FAST-QPA` is called on problem (12) with $\ell^2 = (0,1)^\top$. From Figure 1d we see that `FAST-QPA` stops at iteration $\hat{k}(\ell^2)$ detecting $\lambda^{\ell^2,\hat{k}(\ell^2)}$ such that $\sigma_{\mathcal{L}}(u^3, \lambda^{\ell^2,\hat{k}(\ell^2)}) < 0$. Then, by Lemma 6.1, we have that (Cond) holds and the node can be pruned. Note that `DEIA-BB` did not have to compute LB^r. In particular, we did not need to solve the (duals of) $|L|$ single-objective problems ($P^r(\ell)$) to optimality.

7 Numerical results

In order to investigate the performance of `DEIA-BB`, we considered randomly generated instances of (MOMIQP) with $m \in \{2,3,4\}$. The instances were built with a number of variables $n \in \{5, 10, 15, 20\}$, a number of constraints $p = 15$, and a percentage of integer variables out of the total number of variables equal to %int $\in \{25, 50, 75, 100\}$ (rounded up). Matrices $Q_j \in \mathbb{R}^{n\times n}$, $j \in [m]$ were built using the MATLAB function `sprandsym` and we considered three different density levels $\rho \in \{0.25, 0.50, 0.75\}$. Namely, we generated matrices with approximately $\rho \cdot n^2$ nonzeros entries. For what concerns the linear inequality

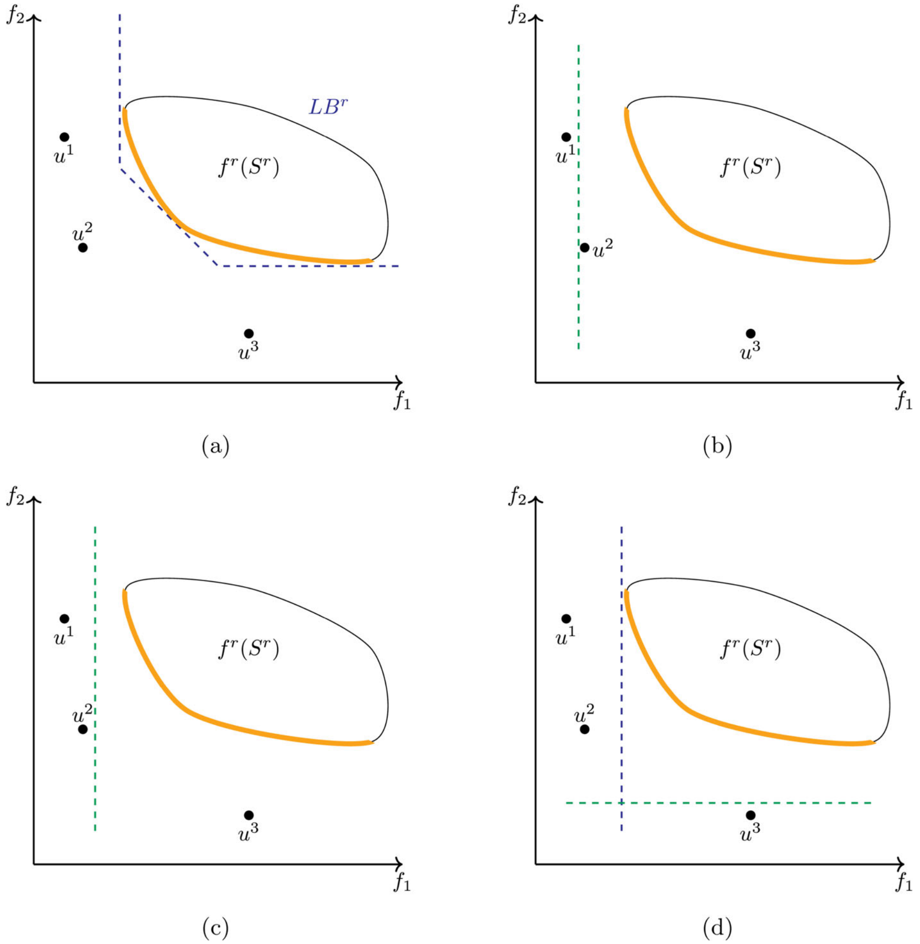

Fig. 1 Outer approximations of $f^r(S^r)$ obtained through dual relaxations - example of *early pruning*

constraints, we randomly generated matrix $A \in \mathbb{R}^{p\times n}$ and vectors $b \in \mathbb{R}^p$ with $m = 15$ using the MATLAB functions `sprandsym` and `rand` respectively. For each combination of $n, m, \%int, \rho$ we produced 5 different instances, having a total of 720 instances. All the algorithms considered have been implemented in MATLAB. In our implementation of `DEIA-BB` we considered $\{1, \ldots, k\}$ as order for fixing. Note that different orderings, as well as alternative branching strategies, could be explored. However, in these numerical tests we focus in analyzing the performance with approximations of the upper image set $\mathcal{P}^r$, obtained from different values for $|L|$ and the related benefits in using dual subproblems. All experiments have been performed on an Intel Core i5-14400F processor running at 2.80GHz under Linux. Each instance was addressed by `DEIA-BB` and, in case the algorithm stopped within one hour, we considered the instance solved by `DEIA-BB` and the sets $\mathcal{L}$ and $\mathcal{U}$ were considered to build an enclosure of the nondominated set.

In Tables 1,2 and 3 we report the results obtained running `DEIA-BB` on instances with $m = 2$, $m = 3$ and $m = 4$, respectively, checking the width of the enclosure obtained when varying the percentage of integer variables. In each table, we report a comparison among three

Table 1 Performance of `DEIA-BB` according to the cardinality of L, $m = 2$

n	%int	$\lvert L\rvert = m$ Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	$\lvert L\rvert = m + 1$ sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	$\lvert L\rvert = m + 1 + m(m - 1)$ Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$
5	25	15	0.01	14.47	0.37	4.47	15	0.01	14.53	0.37	4.47	15	0.01	14.53	0.37	4.47
5	50	15	0.02	32.13	0.18	6.13	15	0.03	31.67	0.18	6.07	15	0.04	31.33	0.18	6.07
5	75	15	0.06	60.00	0.06	6.07	15	0.07	63.20	0.06	6.47	15	0.10	62.53	0.06	6.47
5	100	15	0.10	98.40	0.00	5.00	15	0.12	97.80	0.00	5.00	15	0.16	97.93	0.00	5.00
10	25	15	0.04	100.33	1.97	34.87	15	3.78	83.47	1.97	29.40	15	3.76	80.87	1.97	28.20
10	50	15	0.07	405.73	1.05	78.40	15	3.81	307.27	1.05	59.27	15	3.82	289.87	1.05	55.93
10	75	15	0.39	1486.40	0.42	71.93	15	4.15	1224.93	0.42	65.80	15	4.20	1164.07	0.42	63.33
10	100	15	1.01	2951.27	0.00	48.33	15	4.82	2672.80	0.00	48.87	15	5.05	2617.40	0.00	49.73
15	25	15	9.71	471.07	3.22	182.93	15	9.75	361.00	3.22	138.20	15	9.80	332.60	3.25	125.33
15	50	15	35.45	7245.13	1.60	1192.87	15	23.16	3666.13	1.60	553.80	15	24.96	3187.80	1.62	483.00
15	75	15	33.79	16766.53	0.49	417.27	15	26.28	10228.53	0.49	329.60	15	27.40	9139.60	0.49	301.47
15	100	15	48.82	35666.07	0.00	191.67	15	40.87	26302.53	0.00	196.40	15	40.34	24763.87	0.00	197.53
20	25	15	0.29	2338.80	4.20	870.73	15	0.25	1524.00	4.20	577.00	15	0.31	1397.60	4.20	528.20
20	50	15	593.37	111133.73	2.19	21186.07	15	432.58	32305.20	2.10	5586.13	15	480.05	27332.13	2.11	4778.80
20	75	15	900.83	553223.13	1.17	48487.60	15	684.37	134567.60	1.15	9535.33	15	810.70	112583.07	1.15	8377.93
20	100	15	970.71	579603.87	0.00	723.33	14	780.57	429258.00	0.00	698.93	14	872.44	396691.21	0.00	830.00

Table 2 Performance of `DEIA-BB` according to the cardinality of L, $m = 3$

n	%int	$\lvert L\rvert = m$					$\lvert L\rvert = m + 1$					$\lvert L\rvert = m + 1 + m(m - 1)$				
		Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$
5	25	15	0.02	15.40	0.57	4.93	15	0.01	15.00	0.57	4.93	15	0.02	13.87	0.57	4.67
5	50	15	0.05	38.53	0.41	7.53	15	0.06	38.87	0.41	7.53	15	0.09	36.60	0.41	7.27
5	75	15	0.14	71.40	0.13	6.87	15	0.14	68.20	0.13	6.87	15	0.27	71.80	0.14	7.07
5	100	15	0.23	107.73	0.00	6.93	15	0.25	104.47	0.00	6.87	15	0.43	102.20	0.00	6.80
10	25	15	1.09	117.67	1.89	39.27	15	1.10	117.07	1.89	39.27	15	3.72	88.20	1.89	29.67
10	50	15	1.16	623.27	1.16	105.60	15	1.17	616.60	1.16	103.73	15	3.84	458.40	1.16	77.60
10	75	15	1.07	2378.93	0.53	112.67	15	1.18	2357.87	0.53	112.20	15	4.19	1977.87	0.53	106.73
10	100	15	4.55	5308.33	0.00	91.33	15	4.86	5297.20	0.00	90.87	15	8.73	4780.07	0.00	95.87
15	25	15	1.58	646.00	3.33	253.67	15	1.60	644.07	3.33	251.93	15	6.65	435.60	3.34	166.27
15	50	15	188.14	12864.87	1.89	1948.47	15	237.46	12486.60	1.89	1885.60	15	194.00	5645.47	1.89	866.20
15	75	15	141.90	42518.60	0.68	1334.67	15	189.53	40386.87	0.68	1275.73	15	165.58	21000.20	0.66	911.67
15	100	15	197.10	101983.20	0.00	645.53	15	249.39	98308.60	0.00	644.33	15	217.71	67315.47	0.00	659.20
20	25	15	1.30	5278.87	5.97	2313.07	15	1.40	5166.33	5.97	2257.87	15	1.71	3531.33	5.97	1507.93
20	50	13	1287.60	711606.77	3.59	179597.77	13	1429.31	649473.69	3.55	163624.62	12	1251.97	261886.42	3.40	63594.83
20	75	10	1749.49	2094593.20	1.59	227925.60	11	1868.48	2381734.91	1.69	264657.18	12	1481.86	428816.67	1.79	41700.17
20	100	10	1971.48	2390142.10	0.00	5516.90	11	2176.98	2322670.27	0.00	6353.55	11	1713.09	624516.27	0.00	6031.36

Table 3 Performance of `DEIA-BB` according to the cardinality of L, $m = 4$

n	%int	$\|L\| = m$					$\|L\| = m + 1$					$\|L\| = m + 1 + m(m-1)$				
		Sol	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	*Sol*	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$	*Sol*	Time	Nodes	Width	$\text{Card}_{\mathcal{L}}$
5	25	15	0.02	20.07	0.63	6.33	15	0.03	20.07	0.63	6.33	15	0.05	18.07	0.63	5.87
5	50	15	0.13	50.20	0.48	9.67	15	0.13	50.40	0.48	9.80	15	0.23	46.53	0.48	8.93
5	75	15	0.30	94.00	0.20	11.40	15	0.33	93.80	0.18	11.33	15	0.54	89.87	0.20	11.20
5	100	15	0.51	151.93	0.00	10.47	15	0.57	154.67	0.00	10.47	15	0.94	144.93	0.00	10.07
10	25	15	0.16	118.53	1.93	42.40	15	0.17	118.53	1.93	42.40	15	0.59	96.47	1.93	30.67
10	50	15	0.30	689.87	1.18	129.80	15	0.32	689.07	1.18	129.60	15	0.80	463.27	1.18	91.20
10	75	15	1.81	3055.80	0.54	173.27	15	1.92	3051.40	0.54	173.53	15	3.25	2470.13	0.56	159.73
10	100	15	8.59	7236.40	0.00	136.33	15	8.97	7243.67	0.00	136.47	15	12.99	6696.53	0.00	146.47
15	25	15	16.70	818.20	3.63	322.87	15	16.78	817.80	3.63	322.67	15	38.58	560.47	3.63	217.00
15	50	14	395.61	20490.79	2.08	3605.07	14	397.70	20445.07	2.08	3596.14	14	355.50	9248.79	2.08	1590.86
15	75	14	266.02	71079.36	0.80	2659.14	14	270.52	70546.14	0.80	2634.00	13	342.41	35930.00	0.80	1760.15
15	100	14	413.01	191340.07	0.00	1425.36	14	422.65	190224.00	0.00	1427.14	13	568.62	135069.23	0.00	1599.15
20	25	15	4.10	8033.00	6.24	3874.07	15	4.32	8029.47	6.24	3872.60	15	5.01	5048.27	6.24	2295.67
20	50	9	1826.44	699052.67	3.46	151797.00	9	1862.29	696639.67	3.46	151561.78	11	2163.88	545789.73	4.34	142788.55
20	75	1	2670.72	1058692.00	1.32	77642.00	1	2703.42	1039554.00	1.32	75880.00	3	2874.29	869462.33	1.97	82915.00
20	100	0	–	–	–	–	0	–	–	–	–	3	3021.39	1133625.00	0.00	17408.00

versions of DEIA-BB, where we considered a different number of hyperplanes for building the linear outer approximation of $\mathcal{P}^r$. In particular, we set $|L| \in \{m, m+1, m+1+m*(m-1)\}$, having:

- $|L| = m$, if we consider the m vectors of the standard basis;
- $|L| = m + 1$, if we consider the m vectors of the standard basis plus the vector $\frac{e}{\|e\|_1}$;
- $|L| = m + 1 + m * (m - 1)$, if we consider the m vectors of the standard basis plus the vector $\frac{e}{\|e\|_1}$ and all the possible combination of vectors with two components different from zero, with one component equal to 0.25 and the other one equal to 0.75.

Therefore, for $m = 4$ we consider up to $|L| = 17$, meaning that at every node of DEIA-BB we need to address 17 subproblems to get the approximation of the upper image set $\mathcal{P}^r$.

For each version of DEIA-BB, we report the number of instances solved within the time limit (sol), the average CPU time (in seconds), the average number of nodes, the average width of the enclosure obtained and the average cardinality of the set $\mathcal{L}$. All the averages are taken over the number of instances solved within the time limit among the 15 instances built for fixed n and %int. We can notice that the quality of the enclosure obtained with DEIA-BB improves with the percentage of integer variables considered. Note that the value reported is an average on the quality of the enclosures obtained, on instances with matrices of different density. Therefore, its interpretation is not obvious. However, as expected, the width of the computed enclosure is 0 for purely integer instances. Indeed,DEIA-BB is able to detect the exact nondominated set when dealing with purely integer instances. We can notice that as the cardinality of L increases, the number of nodes that DEIA-BB needs to explore decreases, meaning that the improved quality of the approximation of the upper image set pays off. In some cases, this has a very positive impact: note, for example, that the number of instances solved within the time limit for $n = 20$ and $m = 3$ and $m = 4$ increases with the cardinality of L. However, increasing the cardinality of L does not always correspond to a saving in terms of CPU time. This can be explained by the fact that in every node for which the pruning condition is not satisfied, a larger number of subproblems needs to be solved. The cardinality of the set L has an impact with respect to the cardinality of the lower bound set delivered by DEIA-BB: in general, the higher the cardinality of L the smaller is the cardinality of the lower bound set $\mathcal{L}$ (after reducing it to a stable set). We also notice that the CPU time needed by DEIA-BB strongly increases with the number of variables.

In Figure 2, we report the enclosure produced by DEIA-BB on an instance with $m = 2$, $n = 15$, $\%int = 75$ considering $|L| = m = 2$. The lower and upper bound sets are highlighted. The width of the obtained enclosure is 0.3859 (which is smaller than the average value obtained for all the instances having $n = 15$).

Despite DEIA-BB is able to compute an enclosure of the nondominated set of (MOMIQP), its quality, when dealing with mixed integer instances, cannot be controlled by any of the algorithm's parameters. This makes the comparison of DEIA-BB with other solvers a difficult task. We opted for showing the performance of DEIA-BB on purely integer instances in comparison with the performance of AdEnA, the hybrid decision-criterion space method proposed in [20, Algorithm 3]. For this, we used the code of AdEnA provided on GitHub [27]. By default, AdEnA starts with an enclosure of the nondominated set which is given as a single box and refines it until an enclosure with a prescribed width is computed. Clearly, this width cannot be set to zero, so that a fair comparison is not possible. We chose to set ϵ, the parameter controlling the quality of the enclosure released by AdEnA, equal to 0.1. In Table 4, we show results on instances with n up to 15, as we had numerical issues in calling AdEnA on some instances with $n = 20$ and $m = 3$ and $m = 4$. The results are averaged with respect to ρ, the parameter controlling the density of the matrices in the intances. As already

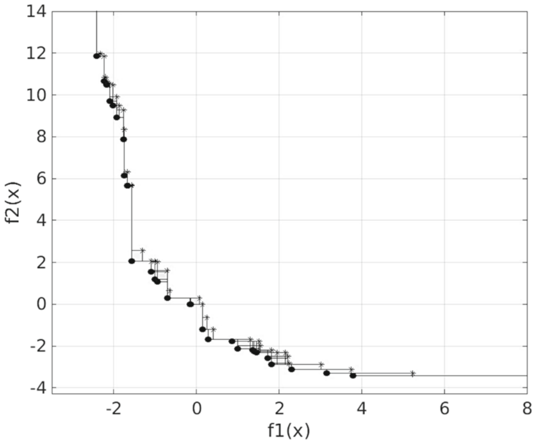

Fig. 2 Enclosure produced by `DEIA-BB` on an instance with $m = 2, n = 15, \%int = 75$

Table 4 Comparison between `DEIA-BB` and `AdEnA` on purely integer instances

	n	ρ	$\|L\| = m$ Sol	Time	$\|L\| = m+1$ Sol	Time	$\|L\| = m+1+m(m-1)$ Sol	Time	`AdEnA` Sol	Time
$m = 2$	5	25	**5**	**0.10**	5	0.12	5	0.15	5	0.15
	5	50	**5**	**0.12**	5	0.15	5	0.19	5	0.15
	5	75	**5**	**0.09**	5	0.11	5	0.14	5	0.17
	10	25	5	0.99	5	1.02	5	1.23	**5**	**0.78**
	10	50	**5**	**0.82**	5	12.20	5	12.43	5	0.92
	10	75	5	1.23	5	1.26	5	1.47	**5**	**0.57**
	15	25	5	70.67	5	78.52	5	77.31	**5**	**2.80**
	15	50	5	43.05	5	20.52	5	22.42	**5**	**3.55**
	15	75	5	32.74	5	23.57	5	21.29	**5**	**4.85**
$m = 3$	5	25	**5**	**0.25**	5	0.24	5	0.43	5	0.39
	5	50	**5**	**0.28**	5	0.32	5	0.52	5	0.73
	5	75	**5**	**0.17**	5	0.20	5	0.35	5	0.51
	10	25	**5**	**4.43**	5	4.74	5	6.22	5	6.99
	10	50	**5**	**3.15**	5	3.35	5	11.77	5	6.77
	10	75	**5**	**6.08**	5	6.48	5	8.21	5	8.63
	15	25	5	62.34	5	116.04	5	142.23	**5**	**38.68**
	15	50	5	154.92	5	246.17	5	300.08	**5**	**40.77**
	15	75	5	374.04	5	385.97	5	210.82	**5**	**77.63**

mentioned `DEIA-BB` particularly suffers from an increasing numer of integer variables, as the number of nodes to be visited strongly grows with respect to this number. However, the time needed by `DEIA-BB` seems to scale better with the number of objective functions than with the number of variables, so that for $m = 4$ the performance of `DEIA-BB` are comparable or even better that those of `AdEnA`.

Table 4 continued

	n	ρ	$\|L\| = m$ Sol	Time	$\|L\| = m+1$ Sol	Time	$\|L\| = m+1+m(m-1)$ Sol	Time	AdEnA Sol	Time
$m = 4$	5	50	**5**	**0.49**	5	0.56	5	0.93	5	1.62
	5	75	**5**	**0.45**	5	0.47	5	0.85	5	1.64
	10	25	**5**	**11.55**	5	11.96	5	17.08	5	32.26
	10	50	**5**	**7.85**	5	8.17	5	11.38	5	39.58
	10	75	**5**	**6.37**	5	6.78	5	10.52	5	31.15
	15	25	4	679.33	4	684.23	4	803.42	**5**	**217.54**
	15	50	5	264.72	5	269.90	5	406.83	**5**	**252.29**
	15	75	5	348.25	5	366.13	4	536.06	**5**	**303.34**

8 Conclusions

We devised a branch-and-bound method able to compute a superset of the set of efficient integer assignments for multiobjective mixed-integer convex quadratic programs. The solution of dual formulations of specific subproblems combined with a corresponding preprocessing phase enables a fast enumeration of the nodes. For all the major results strong convexity of the objective functions is needed as a basic assumption. Already for linear objective functions, i.e., for convex functions which are not strongly convex, the results would fail. This can be seen by the simple example $f : \mathbb{R} \to \mathbb{R}^2$, with $f(x) = (x, -x)$ and feasible set $\mathbb{Z}$. All feasible points are efficient and the nondominated set is unbounded. Also for using the dual relaxations the convexity is needed as a necessary condition for strong duality. Moreover, for explicitly stating the dual problem as done in (12), the analytically given unique solution $x(\lambda)$ is required which is, for nonlinear problems, in general only available for strictly convex quadratic problems. A possible way to make use of those dual bounds for more general nonlinear problems could be to first find strictly convex quadratic underestimators for the objective functions, i.e., strictly convex quadratic functions $g_j : \mathbb{R}^n \to \mathbb{R}$ with $g_j(x) \leq f_j(x)$ for all $x \in \mathbb{R}^n$ for $j = 1, \dots, m$, and then to apply the dual lower bounding procedure on those. That would still give lower bounds for the original functions f_j.

Under the assumption of strong convexity of the objective functions, the algorithm is guaranteed to compute an enclosure of the nondominated set in a finite number of iterations. No assumption on the boundedness of the feasible set is needed. Numerical results on biobjective instances as well as on instances with three and four objectives are reported, showing the ability of the method in computing an enclosure of the nondominated set of multiobjective mixed-integer convex quadratic programs. However, DEIA-BB can be considered exact only in the case of purely integer instances, where the complete nondominated set is delivered. Indeed, for mixed-integer instances the quality of the enclosure obtained cannot be controlled a priori by any of DEIA-BB parameters and from our numerical tests we notice that such quality got worse as the number of continuous variables increases. This weakness of DEIA-BB opens the possibility, as future work, of studying post-processing procedures able to refine the enclosure obtained up to a desired precision.

Appendix

Consider a multiobjective programming problem of the form

$$\begin{array}{ll} \min & (f_1(x), \dots, f_m(x))^\top \\ \text{s.t.} & x \in \Omega \subseteq \mathbb{R}^n \end{array} \tag{15}$$

where $f_j : \mathbb{R}^n \to \mathbb{R}$ denotes a strongly convex function with parameter $\gamma_j > 0$ for all $j \in [m]$ and $\Omega \subseteq \mathbb{R}^n$ is a nonempty feasible set. The following results show that the efficient set and the nondominated set of (15) are bounded sets.

Proposition 1 *Let $\mathcal{E}$ be the efficient set of (15). Then there exist $\bar{x}, \underline{x} \in \mathbb{R}^n$ such that $\mathcal{E} \subseteq$* $\operatorname{int}(B_{\mathcal{E}}) =: (\bar{x}, \underline{x})$.

Proof Assume by contradiction that $(x^k)_{k\in\mathbb{N}} \subseteq \mathcal{E}$ exists such that $\lim_{k\to\infty} \left\|x^k\right\|_2^2 = +\infty$. Let $\tilde{x} \in \Omega$ and let $\rho := f(\tilde{x})$. In particular, we have that

$$\lim_{k\to\infty} \left\|x^k - \tilde{x}\right\|_2^2 = +\infty. \tag{16}$$

Further, let $j \in [m]$. Since f_j is strongly convex, we have that

$$0.25\,\gamma_j\,\|x - \tilde{x}\|_2^2 + f_j(0.5(x + \tilde{x})) \le 0.5 f_j(x) + 0.5 f_j(\tilde{x})$$

holds for all $x \in \mathbb{R}^n$ or, equivalently,

$$0.5\,\gamma_j\,\|x - \tilde{x}\|_2^2 + 2 f_j(0.5(x + \tilde{x})) - f_j(\tilde{x}) \le f_j(x). \tag{17}$$

Furthermore, let x_j^* denote the unique minimizer of f_j over $\mathbb{R}^n$. From (17) and since $f_j(x) \ge f_j(x_j^*)$ for all $x \in \mathbb{R}^n$, we have that for all $k \in \mathbb{N}$

$$f_j(x^k) \ge 0.5\,\gamma_j \left\|x^k - \tilde{x}\right\|_2^2 + 2 f_j(x_j^*) - f_j(\tilde{x}).$$

Therefore, from (16), it necessarily holds $\lim_{k\to\infty} f_j(x^k) = +\infty$ for all $j \in [m]$. In particular, for sufficiently large $k \in \mathbb{N}$ we have that $f(x^k) > \rho = f(\tilde{x})$, contradicting that $(x^k)_{k\in\mathbb{N}} \subseteq \mathcal{E}$. □

As a direct consequence from Proposition 1 we obtain the following:

Corollary 8.1 *Let $\mathcal{N}$ be the nondominated set of (15). Then there exist $z, Z \in \mathbb{R}^m$ such that* $\mathcal{N} \subseteq \operatorname{int}(B_{\mathcal{N}}) =: (z, Z)$.

Acknowledgements The authors are very grateful to the anonymous referees for the careful reading of the manuscript and their valuable comments.

Funding Open access funding provided by Università degli Studi di Firenze within the CRUI-CARE Agreement. The work of the second and the last author is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within Project-ID 432218631. The first and the third author acknowledge support within the project No RM120172A2970290 which has received funding from Sapienza University of Rome.

Data Availability The data presented in this manuscript are reproducible through the implementation publicly available on GitHub https://github.com/mariannadesantis/DEIA-BB

Declarations

Conflict of interest The authors declare they have no financial interests.

References

1. Zhong, H., Guan, W., Zhang, W., Jiang, S., Fan, L.: A bi-objective integer programming model for partly-restricted flight departure scheduling. PLOS ONE **13**(5), 0196146 (2018). https://doi.org/10.1371/journal.pone.0196146
2. Ulusoy, A.-J., Pecci, F., Stoianov, I.: Bi-objective design-for-control of water distribution networks with global bounds. Optim. Eng. **23**(1), 527–577 (2022). https://doi.org/10.1007/s11081-021-09598-z
3. Pecci, F., Stoianov, I.: Bounds and convex heuristics for bi-objective optimal experiment design in water networks. Comput. Oper. Res. **153**, 106181 (2023). https://doi.org/10.1016/j.cor.2023.106181
4. Legendre, A., Angel, E., Tahi, F.: Bi-objective integer programming for RNA secondary structure prediction with pseudoknots. BMC Bioinf. **19**(1), 13 (2018). https://doi.org/10.1186/s12859-018-2007-7
5. Ruzika, S., Wiecek, M.M.: Approximation methods in multiobjective programming. J. Optim. Theory Appl. **126**(3), 473–501 (2005). https://doi.org/10.1007/s10957-005-5494-4
6. Eichfelder, G., Kirst, P., Meng, L., Stein, O.: A general branch-and-bound framework for continuous global multiobjective optimization. J. Global Optim. **80**(1), 195–227 (2021). https://doi.org/10.1007/s10898-020-00984-y
7. Eichfelder, G., Warnow, L.: An approximation algorithm for multi-objective optimization problems using a box-coverage. J. Global Optim. **83**(2), 329–357 (2022). https://doi.org/10.1007/s10898-021-01109-9
8. Eichfelder, G., Stein, O., Warnow, L.: A solver for multiobjective mixed-integer convex and nonconvex optimization. J. Optim. Theory Appl. (2023). https://doi.org/10.1007/s10957-023-02285-2
9. Eichfelder, G., Warnow, L.: A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. Math. Methods Oper. Res. (2023). https://doi.org/10.1007/s00186-023-00828-x
10. De Santis, M., Grani, G., Palagi, L.: Branching with hyperplanes in the criterion space: the frontier partitioner algorithm for biobjective integer programming. European J. Oper. Res. **283**(1), 57–69 (2020). https://doi.org/10.1016/j.ejor.2019.10.034
11. Forget, N., Gadegaard, S.L., Klamroth, K., Nielsen, L.R., Przybylski, A.: Branch-and-bound and objective branching with three or more objectives. Comput. Oper. Res. **148**, 106012 (2022). https://doi.org/10.1016/j.cor.2022.106012
12. Parragh, S.N., Tricoire, F.: Branch-and-bound for bi-objective integer programming. Inf. J. Comput. **31**(4), 805–822 (2019). https://doi.org/10.1287/ijoc.2018.0856
13. De Santis, M., Eichfelder, G., Niebling, J., Rocktäschel, S.: Solving multiobjective mixed integer convex optimization problems. SIAM J. Optim. **30**(4), 3122–3145 (2020). https://doi.org/10.1137/19m1264709
14. Jayasekara Merenchige, P.L.W., Wiecek, M.: A branch and bound algorithm for biobjective mixed integer quadratic programs. Preprint 21294, Optimization Online (2022)
15. Przybylski, A., Gandibleux, X.: Multi-objective branch and bound. Eur. J. Oper. Res. **260**(3), 856–872 (2017). https://doi.org/10.1016/j.ejor.2017.01.032
16. Halffmann, P., Schäfer, L.E., Dächert, K., Klamroth, K., Ruzika, S.: Exact algorithms for multiobjective linear optimization problems with integer variables: A state of the art survey. J. Multi-Criteria Decis. Anal. **29**(5–6), 341–363 (2022). https://doi.org/10.1002/mcda.1780
17. Cabrera-Guerrero, G., Ehrgott, M., Mason, A.J., Raith, A.: Bi-objective optimisation over a set of convex sub-problems. Ann. Oper. Res. **319**(2), 1507–1532 (2022). https://doi.org/10.1007/s10479-020-03910-3
18. De Santis, M., Eichfelder, G.: A decision space algorithm for multiobjective convex quadratic integer optimization. Comput. Oper. Res. **134**, 105396 (2021). https://doi.org/10.1016/j.cor.2021.105396
19. Ehrgott, M.: Multicriteria Optimization. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-27659-9

20. Eichfelder, G., Warnow, L.: Advancements in the computation of enclosures for multi-objective optimization problems. Eur. J. Oper. Res. **310**(1), 315–327 (2023). https://doi.org/10.1016/j.ejor.2023.02.032
21. Klamroth, K., Lacour, R., Vanderpooten, D.: On the representation of the search region in multi-objective optimization. Eur. J. Oper. Res. **245**(3), 767–778 (2015). https://doi.org/10.1016/j.ejor.2015.03.031
22. Buchheim, C., De Santis, M., Palagi, L.: A fast branch-and-bound algorithm for non-convex quadratic integer optimization subject to linear constraints using ellipsoidal relaxations. Oper. Res. Lett. **43**(4), 384–388 (2015). https://doi.org/10.1016/j.orl.2015.05.001
23. Buchheim, C., De Santis, M., Palagi, L., Piacentini, M.: An exact algorithm for nonconvex quadratic integer minimization using ellipsoidal relaxations. SIAM J. Optim. **23**(3), 1867–1889 (2013). https://doi.org/10.1137/120878495
24. Buchheim, C., Caprara, A., Lodi, A.: An effective branch-and-bound algorithm for convex quadratic integer programming. Math. Program. **135**(1–2), 369–395 (2012). https://doi.org/10.1007/s10107-011-0475-x
25. Buchheim, C., De Santis, M., Lucidi, S., Rinaldi, F., Trieu, L.: A feasible active set method with reoptimization for convex quadratic mixed-integer programming. SIAM J. Optim. **26**(3), 1695–1714 (2016). https://doi.org/10.1137/140978971
26. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999). https://doi.org/10.1007/978-0-387-40065-5
27. Eichfelder, G., Warnow, L.: AdEnA. https://github.com/LeoWarnow/AdEnA. Accessed 2023-09-20 (2023)