Quadratic Convex Reformulations for MultiObjective Binary Quadratic Programming

Marianna De Santis* Lucas Létocart† Yue Zhang†

Abstract

Multiobjective binary quadratic programming refers to optimization problems involving multiple quadratic - potentially non-convex - objective functions and a feasible set that includes binary constraints on the variables. In this paper, we extend the well-established Quadratic Convex Reformulation technique, originally developed for single-objective binary quadratic programs, to the multiobjective setting. We propose a branch-and-bound algorithm where lower bound sets are derived from properly defined quadratic convex subproblems. Computational experiments on multiobjective k-item Quadratic Knapsack and multiobjective Max-Cut instances demonstrate the effectiveness of our approach.

Key Words: Multiobjective Optimization, Binary Quadratic Problems, Quadratic Convex Reformulations, Branch-and-Bound algorithm.

Mathematics subject classifications (MSC 2020): 90C11, 90C29, 90C57.

1 Introduction

Multiobjective mixed-integer nonlinear programming refers to a class of mathematical problems where multiple, conflicting nonlinear objective functions need to be optimized over a feasible set that includes integrality constraints on the variables. These problems have numerous applications in applied sciences and engineering. In particular, the growing emphasis on sustainability across various sectors has led to modeling real decision problems in multiobjective terms. Notable applications include renewable energy storage systems design, transportation planning, water distribution network design, and biological studies [1, 35, 34, 32, 25]. When addressing multiobjective mixed integer nonlinear programming problems, a typical goal is to compute an approximation of the nondominated set, and the majority of recently devised approaches with correctness guarantees rely on the concept of *enclosure* (see e.g. [19, 20, 28, 11, 16]), i.e., unions of boxes containing the nondominated set (see [15, 18] for a detailed definition of enclosures in multiobjective problems). On the other hand, when dealing with purely integer problems, the goal is to exactly detect the nondominated set, that is a finite set. Various methods with

*Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze, Via di Santa Marta 3, 50139 Firenze, Italia (`marianna.desantis@unifi.it`)

†Laboratoire d'Informatique de l'Université Paris Nord, Université Sorbonne Paris Nord, 99 Av. Jean Baptiste Clément, 93430 Villetaneuse, France (`lucas.letocart@lipn.univ-paris13.fr`) (`yue.zhang@lipn.univ-paris13.fr`)

correctness guarantees proposed in the literature are branch-and-bound frameworks. We mention, for instance, [12, 9, 21, 31] for multiobjective integer programming and [10, 17, 23, 30] for multiobjective mixed integer programming. For a broader survey of branch-and-bound methods, mainly for multiobjective mixed-integer linear programs, we refer to [33]. The survey in [22] extends this collection by also including approaches that do not use a branch-and-bound framework.

This paper aims to focus on multiobjective binary quadratic problems (MOBQPs) and devise exact solution approaches for this class of mathematical programs. We focus on minimizing $p \geq 2$ quadratic, not necessarily convex objective functions over a linear compact polyhedron and binary variables:

$$\begin{aligned} \min_{x} \quad & (f_1(x), \ldots, f_p(x))^\top \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0,1\} \qquad i \in \{1, \ldots, n\}. \end{aligned} \tag{MO-BQP}$$

The objective functions we consider are given by $f_j \colon \mathbb{R}^n \to \mathbb{R}$,

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

for all $j \in [p] := \{1, \ldots, p\}$ with symmetric matrices $Q_1, \ldots, Q_p \in \mathcal{S}^n$, vectors $c^1, \ldots, c^p \in \mathbb{R}^n$. We underline that Problem (MO-BQP) includes multiobjective combinatorial problems such as multiobjective quadratic knapsack or multiobjective Max-Cut. In this paper, we explore how to extend ideas adopted in the Quadratic Convex Reformulation (QCR) methods for single-objective binary quadratic problems [3, 4, 5] to solve (MO-BQP). In this respect, we devise a branch-and-bound framework, named `MObBQ`, that is presented in Section 3. Our branch-and-bound works refining an upper bound set and is able to deliver the finite nondominated outcome. Details on the computation of the upper bound set are given in Section 3.1. In Section 3.2, the use of the quadratic convex reformulation method for building valid lower bound sets is presented, while the correctness of `MObBQ`, i.e. its ability to exactly detect the nondominated set, is shown in Section 3.3. In Section 4, we present numerical results on two combinatorial instances of multiobjective binary quadratic problems: the multiobjective k-item Quadratic Knapsack Problem (MO-kQKP) and the multiobjective Max-Cut Problem (MO-MCP). For bi-objective instances of these two problems, we also report a comparison with the ϵ-constraint [8, 29] method. Some conclusions are summarized in Section 5.

2 Notation and preliminaries

For an introduction to multiobjective optimization we refer to [13]. We use the standard optimality notion based on the componentwise partial ordering in the image space. A feasible point $\bar{x}$ for (MO-BQP) is called *efficient* if there is no feasible point x 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 (MO-BQP) is called *nondominated*, and the image set of all efficient points is denoted as the nondominated set $\mathcal{N}$ (also known, specifically for $p = 2$, as Pareto front). In the following, we denote by $e_j \in \mathbb{R}^p$ the j-th unit vector of $\mathbb{R}^p$. A stable set $S \subseteq \mathbb{R}^p$ is any set of $\mathbb{R}^p$ where for any two points $y^1, y^2 \in S$ with $y^1 \neq y^2$ it holds that $y^1 \not\leq y^2$, i.e., all elements of S are pairwise non-comparable.

2.1 Quadratic convex reformulation method

Here we give a short introduction to the Quadratic Convex Reformulation (QCR) method [3, 5, 4]. Given a single-objective binary quadratic problem

$$\begin{aligned} \min_x \quad & q(x) = x^\top Q x + c^\top x \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0,1\} \qquad\qquad i \in \{1,\dots,n\}, \end{aligned} \tag{BQP}$$

we have that a feasible point x always satisfies

$$x_i^2 = x_i, \qquad i \in \{1,\dots,n\},$$

as the variables are binary and - in case a subset of linear equality constraints exists, say $A_= x = b_=$ - it always holds

$$(A_= x - b_=)^2 = 0,$$

where, for a generic vector $v \in \mathbb{R}^m$, we use the notation $v^2 = v^\top v$. Introducing the parameters $\delta \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$, it is possible to define the function $q_{\delta,\beta} : \mathbb{R}^n \to \mathbb{R}$ as

$$q_{\delta,\beta}(x) = x^\top Q x + c^\top x + \sum_{i=1}^{n} \delta_i (x_i^2 - x_i) + \beta (A_= x - b_=)^2$$

and problem (BQP) can be reformulated as

$$\begin{aligned} \min_x \quad & q_{\delta,\beta}(x) \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0,1\} \quad i \in \{1,\dots,n\}. \end{aligned} \tag{QCR$_{(\delta,\beta)}$}$$

Note that $q_{\delta,\beta}(x)$ is a quadratic function where the quadratic term depends on the parameters $(\delta,\beta) \in \mathbb{R}^{n+1}$ and is defined by the matrix

$$Q_{\delta,\beta} = Q + diag(\delta) + \beta A_=^\top A_=,$$

where with $diag(\delta)$ we denote the diagonal matrix having δ in the diagonal. In particular, since (QCR$_{(\delta,\beta)}$) is a reformulation of (BQP), any lower bound on the optimal value of (QCR$_{(\delta,\beta)}$), obtained for any $(\delta,\beta) \in \mathbb{R}^{n+1}$, is a valid lower bound on the optimal value of (BQP). The ingenious idea proposed in [3, 4, 5] is to find values for the parameters $(\delta,\beta) \in \mathbb{R}^{n+1}$ that ensure the function $q_{\delta,\beta}$ to be convex while also making the bound

$$\theta(\delta,\beta) := \min\{q_{\delta,\beta}(x) \mid Ax \le b,\ x \in [0,1]^n\}$$

as tight as possible. This is achieved by addressing the following problem

$$\max_{\substack{(\delta,\beta)\in\mathbb{R}^{n+1} \\ Q_{\delta,\beta}\succeq 0}} \theta(\delta,\beta) \tag{1}$$

that can be formulated as a semidefinite program, reported in the Appendix for completeness. Given $(\delta^*, \beta^*) \in \mathbb{R}^{n+1}$ optimal solution of (1), we have that problem $(\mathrm{QCR}_{(\delta^*,\beta^*)})$ is a quadratic convex reformulation of (BQP), that can be directly solved via existing solvers for (mixed) integer convex quadratic programs, like CPLEX or GUROBI. It is also possible to define an *unconstrained* quadratic convex reformulation of Problem (BQP), ignoring the linear equality constraints $A_= x = b_=$. More precisely, we can define the function $q_\delta : \mathbb{R}^n \in \mathbb{R}$ as

$$q_\delta(x) := x^\top Q x + c^\top x + \sum_{i=1}^{n} \delta_i (x_i^2 - x_i)$$

and reformulate problem (BQP) as

$$\begin{aligned} \min_x \quad & q_\delta(x) \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0,1\} \quad i \in \{1,\dots,n\}. \end{aligned} \tag{UQCR_δ}$$

Again, solving a specific semidefinite program without taking into account the constraints $Ax \le b$, the optimal value $\delta^{u,*} \in \mathbb{R}^n$ can be obtained, so that $q_{\delta^{u,*}}$ is convex and

$$\theta(\delta^{u,*}) = \max_{\substack{\delta \in \mathbb{R}^n \\ Q_\delta \succeq 0}} \theta(\delta) = \max_{\substack{\delta \in \mathbb{R}^n \\ Q_\delta \succeq 0}} \min_{\substack{Ax \le b \\ x \in [0,1]^n}} q_\delta(x),$$

where $Q_\delta = Q + diag(\delta)$.

Remark 2.1. *Let $\mathcal{F} \subseteq \{0,1\}^n$ be the feasible set of* (BQP) *and let $x^*, x_u^* \in \mathcal{F}$ be such that*

$$\theta(\delta^*, \beta^*) = q_{(\delta^*,\beta^*)}(x^*), \qquad \theta(\delta^{u,*}) = q_{\delta^{u,*}}(x_u^*).$$

Then, it holds [5]

$$\theta(\delta^{u,*}) \le \theta(\delta^*, \beta^*), \tag{2}$$

and, for any $\delta \in \mathbb{R}^n$,

$$q_\delta(x_u^*) \le q_{\delta^{u,*}}(x_u^*), \qquad q_\delta(x^*) \le q_{(\delta^*,\beta^*)}(x^*). \tag{3}$$

3 MObBQ: a branch-and-bound framework for MultiObjective Binary Quadratic programs

In this section, we outline our branch-and-bound framework. A scheme of the method is reported in Algorithm 1. The search tree is built by branching on the binary variables: if a node is not pruned, a free variable $x_i \in \{0,1\}$ is selected and two subproblems are generated by fixing $x_i = 0$ and $x_i = 1$, respectively. Then, the subproblem at each node of our search tree is defined according to a vector of fixings $r_d \in \{0,1\}^d$, whose dimension d specifies the level of the node in the search tree. Assuming, without loss of generality, that the fixed variables are the first d variables, the subproblem at the node N^{r_d} associated with $r_d \in \{0,1\}^d$ takes the following form:

$$\begin{aligned} \min_x \quad & (f_1(x), \dots, f_p(x))^\top \\ \text{s.t.} \quad & Ax \le b \\ & x_i = r_i \qquad i \in \{1,\dots,d\}, \\ & x_i \in \{0,1\} \qquad i \in \{d+1,\dots,n\}. \end{aligned} \tag{4}$$

Note that such a problem is still a multiobjective binary quadratic problem. By explicitly incorporating the fixings on the integer variables, we can reduce its dimension to $n - d$ (see Section 3.2).

As for the majority of branch-and-bound methods for multiobjective (mixed) integer problems (see, e.g., [10, 17, 18, 20]), our approach is essentially based on evaluating lower bound and upper bound sets to decide whether a node should be pruned. Details are given in the next sections. The iterative routine for subproblems is first to build a lower bound set by solving a properly defined relaxation (line 8). The global upper bound set is updated whenever a new feasible point is encountered. A node is pruned by infeasibility, integrity or dominance, otherwise new nodes are created (lines 15-16).

Algorithm 1 `MObBQ`

INPUT: (MO-BQP)
OUTPUT: $S = \mathcal{N} \subseteq \mathbb{R}^p$, the finite nondominated set of (MO-BQP)
1: Compute a starting finite and stable set S of images of feasible points of (MO-BQP)
2: Compute the upper bound set $\mathcal{U} = U(S)$
3: Set $d = 0$, $r_d = \emptyset$
4: $\mathcal{L} := \{N^{r_d}\}$
5: **while** $\mathcal{L} \neq \emptyset$ **do**
6:     Choose $N^{r_d} \in \mathcal{L}$
7:     $\mathcal{L} \leftarrow \mathcal{L} \setminus \{N^{r_d}\}$
8:     Compute a lower bound set LB^{r_d} and evaluate `Prune(`N^{r_d}`)`
9:     **if** `Prune(`N^{r_d}`)` **then**
10:         Go to 6
11:     **else if** `CheckInt(`N^{r_d}`)` **then**
12:         Update $\mathcal{U} = U(S \cup f(\bar{x}))$
13:         Update S including $f(\bar{x})$ and keeping S a stable set
14:     **else**
15:         Set $r^0_{d+1} = (r_d, 0) \in \{0,1\}^{d+1}$, $r^1_{d+1} = (r_d, 1) \in \{0,1\}^{d+1}$
16:         $\mathcal{L} \leftarrow \mathcal{L} \cup \{N^{r^0_{d+1}}, N^{r^1_{d+1}}\}$,
17:     **end if**
18: **end while**

3.1 Computation of upper bound sets and local upper bounds

An upper bound set of the nondominated set $\mathcal{N}$ of (MO-BQP) is any set $U \subseteq \mathbb{R}^p$ that satisfies $\mathcal{N} \subseteq U - \mathbb{R}^p_+$. One way to obtain such a set is to rely on the so-called *local upper bounds* and the related concept of *search zones*, introduced in [24].

In the following, we adopt the generalized definition of lower search zones and local upper bounds from [19, Def. 4.1].

Starting from an *initial area of interest* B, i.e. an arbitrary subset of $\mathbb{R}^p$ with $\mathcal{N} \subseteq \text{int}(B)$ (see [19, Assumption 4.3]), we can define the *lower search zone* of a point and of a finite and stable set in the criterion space as follows.

Definition 3.1. *Let $B \subseteq \mathbb{R}^p$ with $\mathcal{N} \subseteq \text{int}(B)$ and let $u \in \mathbb{R}^p$. The lower search zone defined by u is*

$$c(u) := \{y \in \text{int}(B) \mid y < u\}.$$

Given a finite and stable set $S \subseteq \mathbb{R}^p$, the lower search region for S is

$$sr(S) := \text{int}(B) \setminus (S + \mathbb{R}^p_+).$$

Note that the nondominated set $\mathcal{N}$ of (MO-BQP) is a finite set in $\mathbb{R}^p$, so that the area of interest B can be chosen as a box, i.e. $\mathcal{N} \subseteq int(B) = (z, Z) := \{x \in \mathbb{R}^p \mid z < x < Z\}$, with $z, Z \in \mathbb{R}^p$. Starting from the definition of lower search zones for points in the criterion space, we can define the *upper bound set* for a finite and stable subset of $\mathbb{R}^p$.

Definition 3.2. *Given a finite and stable set $S \subseteq \mathbb{R}^p$, the set $\mathcal{U} = U(S) \subseteq \mathbb{R}^p$ is the upper bound set related to S if*

i. $sr(S) = \bigcup_{u \in U(S)} c(u)$

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

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

In our branch-and-bound method, we start with an initial finite and stable set $S \subseteq \mathbb{R}^p$. Such set can be the empty set or can be initialized by computing feasible points for (MO-BQP) using an appropriate heuristic.Then, we incrementally include images of feasible points obtained along the enumeration of the nodes, ensuring that S remains a stable set. Once the list of nodes to be explored is empty, the set S is the nondominated set $\mathcal{N}$ of (MO-BQP) (see Algorithm 1 and Section 3.3).

More precisely, given the finite and stable set $S \subseteq \mathbb{R}^p$ and a feasible point $\hat{x} \in \{0,1\}^n$ of (MO-BQP), the upper bound set $\mathcal{U}$ is updated according to the procedure [24, Algorithm 3]. In particular, every time a leaf node of the search tree leads to a feasible point $r_d \in \{0,1\}^n$ of (MO-BQP) the upper bound set is updated as

$$\mathcal{U} = U(S \cup \{f(r_d)\})$$

using the procedure [24, Algorithm 3] and $f(r_d)$ is added to S, keeping S a stable set. As shown in [24], for the resulting upper bound set it holds that

$$\mathcal{N} \subseteq \mathcal{U} - \mathbb{R}^p_+. \tag{5}$$

3.2 Lower bound sets through quadratic convex reformulations

In this section, we explore how to transfer the quadratic convex reformulation paradigm to deal with (MO-BQP). Let $\delta^{u,*}_{0,j} \in \mathbb{R}^n$ and $(\delta^*_{0,j}, \beta^*_{0,j}) \in \mathbb{R}^{n+1}$ be the optimal parameters for the unconstrained and constrained quadratic convex reformulation method obtained for the objective function $f_j(x)$, $j \in [p]$ over the feasible set of (MO-BQP), respectively. From what reported in Section 2.1, we have that both

$$(q_j)_{\delta^{u,*}_{0,j}}(x) := f_j(x) + \sum_{i=1}^{n} (\delta^{u,*}_{0,j})_i (x_i^2 - x_i)$$

and

$$(q_j)_{(\delta^*_{0,j}, \beta^*_{0,j})}(x) := f_j(x) + \sum_{i=1}^{n} (\delta^*_{0,j})_i (x_i^2 - x_i) + \beta^*_{0,j}(A_= x - b_=)^2$$

are convex quadratic objective functions. Using these functions, we define the following multi-objective convex binary quadratic problems:

$$\begin{aligned} \min_x \quad & ((q_1)_{\delta^{u,*}_{0,1}}(x), \ldots, (q_p)_{\delta^{u,*}_{0,p}}(x))^\top \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0,1\} \qquad\qquad i \in [n], \end{aligned} \tag{MO-UQCR}$$

$$\begin{aligned} \min_x \quad & ((q_1)_{(\delta^*_{0,1},\beta^*_{0,1})}(x), \ldots, (q_p)_{(\delta^*_{0,p},\beta^*_{0,p})}(x))^\top \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0,1\} \qquad\qquad i \in [n]. \end{aligned} \tag{MO-QCR}$$

We can easily show that (MO-UQCR) and (MO-QCR) are valid reformulations of (MO-BQP).

Proposition 3.3. *Let* $\mathcal{N}$*,* $\mathcal{N}_{UQCR}$ *and* $\mathcal{N}_{QCR}$ *be the nondominated sets of* (MO-BQP)*,* (MO-UQCR) *and* (MO-QCR)*, respectively. It holds*

$$\mathcal{N} = \mathcal{N}_{UQCR} = \mathcal{N}_{QCR}.$$

Proof. Let $x \in \{0,1\}^n$ be any feasible point of (MO-BQP). Since $x_i^2 - x_i = 0$, for all $i \in [n]$ and $A_= x = b_=$, then

$$(q_j)_{\delta^{u,*}_{0,j}}(x) = (q_j)_{(\delta^*_{0,j},\beta^*_{0,j})}(x) = f_j(x), \quad \text{for all } j \in [p].$$

This implies that (MO-BQP), (MO-UQCR) and (MO-QCR) share the same feasible set in the image space and the result holds. □

Note that both (MO-UQCR) and (MO-QCR) are obtained after solving p semidefinite programming problems. Thanks to Proposition 3.3, a possibility for solving (MO-BQP) can be that of directly addressing (MO-UQCR) or (MO-QCR), for which solution approaches exist, see, e.g., [20] and [17]. However, these approaches work by refining *enclosures* of the nondominated set and their output is an enclosure with a prescribed positive width. Therefore, they are not suitable in our context, where the goal is that of detecting the finite nondominated set of (MO-BQP). It is important to note that the branch-and-bound method proposed in [11], which is able to determine the nondominated set for multiobjective integer strongly convex quadratic problems, cannot be directly applied here, as the objective functions in (MO-UQCR) and (MO-QCR) are not guaranteed to be strongly convex. However, as will be clarified below, our branch-and-bound approach operates similarly to the method in [11], as the subproblems in the nodes are defined by fixing integer variables and lower bound sets are obtained as intersections of linear halfspaces.

Given a node N^{r_d}, a lower bound set LB^{r_d} is any set containing the upper image set of the multiobjective binary quadratic subproblem defined at N^{r_d}. Taking explicitly into account the fixings on the variables, the subproblem at node N^{r_d} can be reformulated as

$$\begin{aligned} \min_x \quad & (f_1^{r_d}(x), \ldots, f_p^{r_d}(x))^\top \\ \text{s.t.} \quad & A^d x \le b^{r_d} \\ & x \in \{0,1\}^{n-d}, \end{aligned} \tag{MO-BQPrd}$$

where, for every $j \in [p]$, we define, as in [9, Lemma 3.1], $f_j^{r_d} : \mathbb{R}^{n-d} \to \mathbb{R}$ by $f_j^{r_d}(x) := f_j(r_1, \ldots, r_d, x_1, \ldots, x_{n-d})$. Matrix A^d denotes the matrix obtained from A deleting the first d columns and $b^{r_d} := b - A(r_1, \ldots, r_d, 0, \ldots, 0)^\top$. Each function $f_j^{r_d}$ can be expressed as

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

where the symmetric matrix Q_j^d is obtained by deleting the corresponding d rows and columns of Q_j and c^{j,r_d} and a^{j,r_d} are set to

$$c_{i-d}^{j,r_d} := c_i^j + 2\sum_{l=1}^{d} (Q_j)_{li}(r_d)_l, \quad \text{for} \quad i = d+1, \ldots, n$$

and

$$a^{j,r_d} := \sum_{l=1}^{d} c_l^j (r_d)_l + \sum_{l=1}^{d}\sum_{i=1}^{d} (Q_j)_{li}(r_d)_l(r_d)_i.$$

To compute valid lower bound sets LB^{r_d} of (MO-BQPrd), we rely on both its *unconstrained* quadratic convex reformulation, that uses only the perturbation related to the binary constraints, and its *constrained* quadratic convex reformulation, that exploits also the perturbation on the linear equality constraints.

For each objective function of (MO-BQPrd), we define $(q_j^{r_d})_{\delta_j^u} : \mathbb{R}^{n-d} \to \mathbb{R}, j \in [p]$ as

$$(q_j^{r_d})_{\delta_j^u}(x) := f_j^{r_d}(x) + \sum_{i=1}^{n-d} (\delta_j^u)_i (x_i^2 - x_i),$$

where $\delta_j^u \in \mathbb{R}^{n-d}$ is a vector of UQCR parameters. We further define $(q_j^{r_d})_{(\delta_j,\beta_j)} : \mathbb{R}^{n-d} \to \mathbb{R}$ as follows:

$$(q_j^{r_d})_{(\delta_j,\beta_j)}(x) := (q_j^{r_d})_{\delta_j}(x) + \beta_j (A_=^d x - b_=^{r_d})^2,$$

where $A_=^d x = b_=^{r_d}$ is a subset of equality constraints of $A^d x \leq b^{r_d}$ and $(\delta_j, \beta_j) \in \mathbb{R}^{n-d+1}$ are QCR parameters. Note that, given $\bar{x} \in \{0,1\}^{n-d}$ feasible for (MO-BQPrd), it holds that

$$f_j^{r_d}(\bar{x}) \; = \; (q_j^{r_d})_{(\delta_j^u)}(\bar{x}) \; = \; (q_j^{r_d})_{(\delta_j,\beta_j)}(\bar{x}), \tag{6}$$

for any $\delta_j^u \in \mathbb{R}^{n-d}$, $(\delta_j, \beta_j) \in \mathbb{R}^{n-d+1}$ and each $j \in [p]$. Applying the quadratic convex reformulation method, using $(q_j^{r_d})_{\delta_j^u}$ and $(q_j^{r_d})_{\delta_j,\beta_j}$, respectively, we obtain two multiobjective binary convex quadratic reformulations of problem (MO-BQPrd). In the following, we denote by $\mathcal{F}^{r_d}$ the linear relaxation domain of N^{r_d}

$$\mathcal{F}^{r_d} := \{x \in [0,1]^{n-d} \mid A^d x \leq b^{r_d}\}.$$

For each $j \in [p]$, let $\delta_j^{u,*} \in \mathbb{R}^{n-d}$ be the vector of optimal values for the UQCR parameters $(\delta_j^u)_i$, $i \in [n-d]$, ensuring that the function $(q_j^{r_d})_{\delta_j^{u,*}}$ is convex and such that the value

$$\theta_j^{r_d}(\delta_j^u) = \min\{(q_j^{r_d})_{\delta_j^u}(x) \mid x \in \mathcal{F}^{r_d}\}$$

is maximized with respect to $\delta_j^u \in \mathbb{R}^{n-d}$. In particular,

$$\theta_j^{r_d}(\delta_j^{u,*}) := \max_{\delta_j^u \in \mathbb{R}^{n-d}} \min\{(q_j^{r_d})_{\delta_j^u}(x) \mid x \in \mathcal{F}^{r_d}\}.$$

Then, we define the following problem, having $(q_j^{r_d})_{\delta_j^{u,*}}(x)$, $j \in [p]$ as objective functions:

$$\begin{aligned} \min_x \quad & (q^{r_d})_{\delta^{u,*}}(x) = ((q_1^{r_d})_{\delta_1^{u,*}}(x), \ldots, (q_p^{r_d})_{\delta_p^{u,*}}(x))^\top \\ \text{s.t.} \quad & A^d x \leq b^{r_d} \\ & x \in \{0,1\}^{n-d}. \end{aligned} \tag{MO-UQCRrd}$$

If we also allow the perturbation related to the linear equality constraints, we get the following problem

$$\begin{aligned} \min_x \quad & (q^{r_d})_{(\delta^*,\beta^*)}(x) = ((q_1^{r_d})_{(\delta_1^*,\beta_1^*)}(x), \ldots, (q_p^{r_d})_{(\delta_p^*,\beta_p^*)}(x))^\top \\ \text{s.t.} \quad & A^d x \leq b^{r_d} \\ & x \in \{0,1\}^{n-d}, \end{aligned} \tag{MO-QCRrd}$$

where, for each $j \in [p]$, $(\delta_j^*, \beta_j^*) \in \mathbb{R}^{n-d+1}$ is the vector of optimal values for the QCR parameters (δ_j, β_j), ensuring that the function $(q_j^{r_d})_{(\delta_j^*,\beta_j^*)}$ is convex and such that

$$\theta_j^{r_d}(\delta_j^*, \beta_j^*) := \max_{\delta_j, \beta_j \in \mathbb{R}^{n-d+1}} \min\{(q_j^{r_d})_{(\delta_j,\beta_j)}(x) \mid x \in \mathcal{F}^{r_d}\}.$$

Note that $\delta_j^{u,*} \in \mathbb{R}^{n-d}$ within (MO-UQCRrd) differs in general from $\delta_j^* \in \mathbb{R}^{n-d}$ within (MO-QCRrd), as they are obtained as optimal dual variables of two different semidefinite programming problems (see Appendix). Similarly to Proposition 3.3, we can state the following.

Proposition 3.4. *Let $\mathcal{N}^{r_d}$, $\mathcal{N}^{r_d}_{UQCR}$ and $\mathcal{N}^{r_d}_{QCR}$ be the nondominated sets of* (MO-BQPrd), (MO-UQCRrd) *and* (MO-QCRrd), *respectively. It holds*

$$\mathcal{N}^{r_d} = \mathcal{N}^{r_d}_{UQCR} = \mathcal{N}^{r_d}_{QCR}.$$

Proof. From (6), we have that (MO-BQPrd), (MO-UQCRrd) and (MO-QCRrd) share the same feasible set in the image space and the result holds. □

Thanks to Proposition 3.4, a lower bound set for any of (MO-BQPrd), (MO-UQCRrd) and (MO-QCRrd) also serves as a lower bound set for the others. Since both (MO-UQCRrd) and (MO-QCRrd) have convex quadratic objective functions, we compute lower bound sets using outer approximations of the upper image sets of their continuous relaxations. Given the continuous relaxations of (MO-UQCRrd) and (MO-QCRrd), their upper image sets are denoted as $\mathcal{P}_u^{r_d}$ and $\mathcal{P}^{r_d}$, respectively and defined as

$$\mathcal{P}_u^{r_d} := \{(q^{r_d})_{\delta^{u,*}}(x) \in \mathbb{R}^p \mid x \in \mathcal{F}^{r_d}\} + \mathbb{R}^p_+,$$

and

$$\mathcal{P}^{r_d} := \{(q^{r_d})_{(\delta^*,\beta^*)}(x) \in \mathbb{R}^p \mid x \in \mathcal{F}^{r_d}\} + \mathbb{R}^p_+.$$

In case $\mathcal{F}^{r_d} = \emptyset$ the node N^{r_d} can be pruned as shown in Section 3.3. So, in the following, we assume that $\mathcal{F}^{r_d} \neq \emptyset$.

For both $\mathcal{P}_u^{r_d}$ and $\mathcal{P}^{r_d}$, the simplest outer approximation is the one obtained by p supporting hyperplanes with normal vectors equal to the p unit vectors. This corresponds to compute the ideal points of $\mathcal{P}_u^{r_d}$ and $\mathcal{P}^{r_d}$, which are componentwise defined as $\min\{y_j \in \mathbb{R} \mid y \in \mathcal{P}_u^{r_d}\}$ and $\min\{y_j \in \mathbb{R} \mid y \in \mathcal{P}^{r_d}\}$ for $j \in [p]$, respectively. Note that these minima exist as $\mathcal{F}^{r_d}$ is a nonempty and compact set. The outer approximation obtained from the computation of the ideal point is very rough, therefore we allow to consider further supporting hyperplanes to get better outer approximations as done also in [11].

Let W be a finite set of nonnegative vectors which includes all p unit vectors:

$$\{e_1, \ldots, e_p\} \subseteq W \subseteq \{y \in \mathbb{R}^p_+ \mid \|y\|_1 = 1\}. \tag{7}$$

Given node N^{r_d}, in order to compute an outer approximation for $\mathcal{P}_u^{r_d}$ or $\mathcal{P}^{r_d}$, we solve $|W|$ continuous single-objective subproblems, having as feasible set $\mathcal{F}^{r_d}$ and as objective function the weighted sum with respect to $w \in W$ of $(q^{r_d})_{\delta^{u,*}}(x)$ or $(q^{r_d})_{(\delta^*,\beta^*)}(x)$, respectively. Each optimal solution of the weighted sum subproblem defines a hyperplane used for the outer approximation of $\mathcal{P}_u^{r_d}$ or $\mathcal{P}^{r_d}$.

The idea of including hyperplanes using a weighted sum on a relaxed problem was proposed for multiobjective linear integer problems in [14], see also the survey [33]. In the context of multiobjective convex quadratic problems, this strategy has already been used in [9, 11].

Definition 3.5. *Let W be a finite set as in (7). We define the lower bound sets $LB^{r_d}_{QCR^{r_d}}(W)$ and $LB^{r_d}_{UQCR^{r_d}}(W)$, linear outer approximations of $\mathcal{P}^{r_d}$ and $\mathcal{P}_u^{r_d}$, as follows:*

$$LB^{r_d}_{QCR^{r_d}}(W) := \bigcap_{w \in W} \{y \in \mathbb{R}^p \mid w^\top y \geq \theta^{r_d}(w)\}$$

$$LB^{r_d}_{UQCR^{r_d}}(W) := \bigcap_{w \in W} \{y \in \mathbb{R}^p \mid w^\top y \geq \theta_u^{r_d}(w)\},$$

where, for each $w \in W$, the values $\theta^{r_d}(w)$ and $\theta_u^{r_d}(w)$ are computed as:

$$\theta^{r_d}(w) := \min\{w^\top (q^{r_d})_{(\delta^*,\beta^*)}(x) \mid x \in \mathcal{F}^{r_d}\} \tag{$\mathrm{P}^{rd}(w)$}$$

$$\theta_u^{r_d}(w) := \min\{w^\top (q^{r_d})_{\delta^{u,*}}(x) \mid x \in \mathcal{F}^{r_d}\}. \tag{$\mathrm{P}_u^{rd}(w)$}$$

As a first result, we show that $LB^{r_d}_{QCR^{r_d}}(W)$ is a stronger lower bound set than $LB^{r_d}_{UQCR^{r_d}}(W)$.

Proposition 3.6. *Let N^{r_d} be the node defined according to $r_d \in \{0,1\}^d$. It holds*

$$LB^{r_d}_{QCR^{r_d}}(W) \subseteq LB^{r_d}_{UQCR^{r_d}}(W). \tag{8}$$

Proof. Since $w \in W \subseteq \mathbb{R}^p_+$ we have that showing (8) is equivalent to prove that $\theta_u^{r_d}(w) \leq \theta^{r_d}(w)$ for every $w \in W$. Assume by contradiction that $w \in W$ exists such that

$$\theta_u^{r_d}(w) > \theta^{r_d}(w).$$

Let $\bar{x}, \bar{x}_u \in \mathcal{F}^{r_d}$ be optima for $(\mathrm{P}^{rd}(w))$ and $(\mathrm{P}_u^{rd}(w))$, respectively. In particular, given any $x \in \mathcal{F}^{r_d}$, we can write

$$w^\top (q^{r_d})_{\delta^{u,*}}(x) \geq w^\top (q^{r_d})_{\delta^{u,*}}(\bar{x}_u) = \theta_u^{r_d}(w) > \theta^{r_d}(w) = w^\top (q^{r_d})_{(\delta^*,\beta^*)}(\bar{x}), \tag{9}$$

where the first inequality holds as $\bar{x}_u \in \mathcal{F}^{r_d}$ is optimum for $(\mathrm{P}_u^{rd}(w))$.

Since $w \in W \subseteq \mathbb{R}_+^p$ and $w \neq 0$ as $\|w\|_1 = 1$, (9) implies that, for any $x \in \mathcal{F}^{r_d}$, an index $j \in [p]$ exists such that

$$(q_j^{r_d})_{\delta_j^{u,*}}(x) > (q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(\bar{x}).$$

Let $x^* \in \mathcal{F}^{r_d}$ be an optimum for $(q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(x)$ over $\mathcal{F}^{r_d}$:

$$x^* \in arg\min\{(q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(x) \mid A^d x \leq b^{r_d},\ x \in [0,1]^{n-d}\}.$$

Then, an index $j \in [p]$ exists such that

$$(q_j^{r_d})_{\delta_j^{u,*}}(x^*) > (q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(\bar{x}) \geq (q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(x^*),$$

and we get a contradiction with

$$(q_j^{r_d})_{\delta_j^{u,*}}(x^*) \leq (q_j^{r_d})_{(\delta_j^*,\beta_j^*)}(x^*),$$

where the latter holds from (3) (see Remark 2.1). □

Similarly, we can show that applying the quadratic convex reformulation at each node is better than reformulating the original (MO-BQP) as a convex quadratic problem, as long as we apply the same branch-and-bound scheme reported in Algorithm 1. To be more precise, let $\delta_{0,j}^{u,*} \in \mathbb{R}^n$ and $(\delta_{0,j}^*, \beta_{0,j}^*) \in \mathbb{R}^{n+1}$, $j \in [p]$, be the parameters related to the quadratic convex reformulations of (MO-BQP) presented at the beginning of this section and denoted by (MO-UQCR) and (MO-QCR). We denote by $LB_{QCR^0}^{r_d}(W)$ and by $LB_{UQCR^0}^{r_d}(W)$ the lower bound sets that could be considered at node N^{rd} taking into account (MO-UQCR) and (MO-QCR), namely:

$$LB_{UQCR^0}^{r_d}(W) := \bigcap_{w \in W} \{y \in \mathbb{R}^p \mid w^\top y \geq \theta_{u,0}^{r_d}(w)\}$$

$$LB_{QCR^0}^{r_d}(W) := \bigcap_{w \in W} \{y \in \mathbb{R}^p \mid w^\top y \geq \theta_0^{r_d}(w)\},$$

where, for each $w \in W$, the values $\theta_0^{r_d}(w)$ and $\theta_{u,0}^{r_d}(w)$ are computed as:

$$\theta_{u,0}^{r_d}(w) := \min\{w^\top (q^{r_d})_{\delta_0^{u,*}}(x) \mid x \in \mathcal{F}^{r_d}\} \tag{10}$$

$$\theta_0^{r_d}(w) := \min\{w^\top (q^{r_d})_{(\delta_0^*,\beta_0^*)}(x) \mid x \in \mathcal{F}^{r_d}\}. \tag{11}$$

We can prove that the lower bound sets obtained as in Definition 3.5 are stronger than $LB_{UQCR^0}^{r_d}(W)$ and $LB_{QCR^0}^{r_d}(W)$, respectively:

Proposition 3.7. *Let N^{r_d} be the node defined according to $r_d \in \{0,1\}^d$. It holds*

$$LB_{UQCR^{r_d}}^{r_d}(W) \subseteq LB_{UQCR^0}^{r_d}(W) \quad \textit{and} \quad LB_{QCR^{r_d}}^{r_d}(W) \subseteq LB_{QCR^0}^{r_d}(W).$$

Proof. We only show that $LB^{r_d}_{UQCR^{r_d}}(W) \subseteq LB^{r_d}_{UQCR^0}(W)$ as proving the other inclusion uses the same steps. As in the proof of Proposition 3.6, since $w \in W \subseteq \mathbb{R}^p_+$, we can prove the result by equivalently showing that $\theta^{r_d}_{u,0}(w) \leq \theta^{r_d}_u(w)$ for every $w \in W$. Let $\bar{x}_u, \bar{x}_{u,0} \in \mathcal{F}^{r_d}$ be such that

$$\theta^{r_d}_u(w) = w^\top (q^{r_d})_{\delta^{u,*}}(\bar{x}_u), \qquad \theta^{r_d}_{u,0}(w) = w^\top (q^{r_d})_{\delta^{u,*}_0}(\bar{x}_{u,0}).$$

Assume by contradiction that $w \in W$ exists such that

$$\theta^{r_d}_{u,0}(w) > \theta^{r_d}_u(w).$$

In particular, given any $x \in \mathcal{F}^{r_d}$, we can write

$$w^\top (q^{r_d})_{\delta^{u,*}_0}(x) \geq w^\top (q^{r_d})_{\delta^{u,*}_0}(\bar{x}_{u,0}) = \theta^{r_d}_{u,0}(w) > \theta^{r_d}_u(w) = w^\top (q^{r_d})_{\delta^{u,*}}(\bar{x}_u),$$

where the first inequality holds as $\bar{x}_{u,0} \in \mathcal{F}^{r_d}$ is optimum for (10).

Since $w \in W \subseteq \mathbb{R}^p_+$ and $w \neq 0$ as $\|w\|_1 = 1$, for any $x \in \mathcal{F}^{r_d}$, an index $j \in [p]$ exists such that

$$(q^{r_d}_j)_{\delta^{u,*}_{0,j}}(x) > (q^{r_d}_j)_{\delta^{u,*}_j}(\bar{x}_u).$$

Let $x^*_u \in [0,1]^{n-d}$ be optimum for $(q^{r_d}_j)_{\delta^{u,*}_j}(x)$ over $\mathcal{F}^{r_d}$, i.e.

$$x^*_u \in arg\min\{(q^{r_d}_j)_{\delta^{u,*}_j}(x) \mid A^d x \leq b^{r_d},\ x \in [0,1]^{n-d}\}.$$

Then, an index $j \in [p]$ exists such that

$$(q^{r_d}_j)_{\delta^{u,*}_{0,j}}(x^*_u) > (q^{r_d}_j)_{\delta^{u,*}_j}(\bar{x}_u) \geq (q^{r_d}_j)_{\delta^{u,*}_j}(x^*_u),$$

and we get a contradiction with

$$(q^{r_d}_j)_{\delta^{u,*}_{0,j}}(x^*_u) \leq (q^{r_d}_j)_{\delta^{u,*}_j}(x^*_u),$$

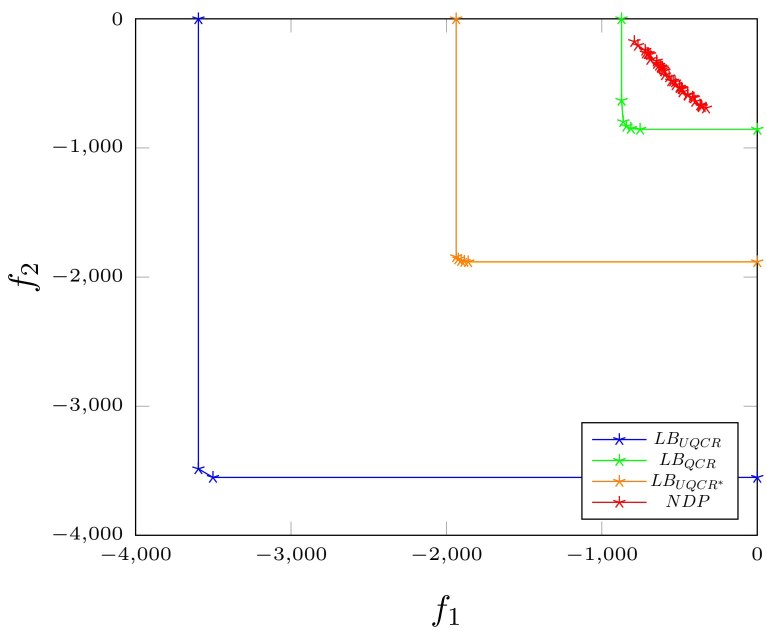
implied by (3) (see Remark 2.1). □

Example 3.8. *Consider the following bi-objective k-item knapsack instance with 20 binary variables*

$$\begin{aligned}
\min_x \quad & -x^T Q_1 x,\ -x^T Q_2 x \\
\text{s.t.} \quad & 47x_1 + 27x_2 + 45x_3 + 50x_4 + 33x_5 + 19x_6 + 41x_7 + 19x_8 + \\
& 39x_9 + 46x_{10} + 7x_{11} + 47x_{12} + 20x_{13} + 21x_{14} + 25x_{15} + 5x_{16} + \\
& 14x_{17} + 34x_{18} + 44x_{19} + 12x_{20} \leq 64 \\
& \sum_{i=1}^{20} x_i = 4 \\
& x \in \{0,1\}^{20},
\end{aligned}$$

with Q_1 and Q_2 being two symmetric full density matrices randomly generated. In Figure 1 we report the set of nondominated points (NDP) of the instance together with the different lower bound sets computed at the root node. It is evident that $LB_{QCR} \subseteq LB_{UQCR}$ as proved in Proposition 3.6. In the picture, we also report the lower bound set obtained from the improved *unconstrained quadratic convex reformulation LB_{UQCR^*} (see Section 4.1), for which is clear that $LB_{QCR} \subseteq LB_{UQCR^*} \subseteq LB_{UQCR}$.*

Figure 1: A comparison on lower bound sets from different quadratic convex reformulations of a bi-objective k-item quadratic knapsack instance.



3.3 Pruning conditions and correctness of `MObBQ`

To prove that `MObBQ` correctly delivers the finite nondominated set of Problem (MO-BQP), we show that a node is pruned only in case it cannot lead to efficient binary assignments, i.e. to fixings of the binary variables that lead to efficient points. As a first result, we state that in case a node N^{r_d} corresponds to an infeasible subproblem, this node and all its children can of course be pruned. Indeed, if $\mathcal{F}^{r_d} = \emptyset$ then (MO-BQP^{r_d}) is infeasible and then r_d is an infeasible fixing:

Lemma 3.9. *Let N^{r_d} be a node such that $\mathcal{F}^{r_d} = \emptyset$. Then there is no feasible point $\bar{x} \in \{0,1\}^n$ of* (MO-BQP) *such that $(\bar{x}_1, \dots, \bar{x}_d) = (r_1, \dots, r_d)$.*

As a second result, that also gives a condition for pruning a node N^{r_d}, we consider the intersection between the set of local upper bounds $\mathcal{U}$ with a lower bound set. Let LB^{r_d} denote a lower bound set at node N^{r_d}. According to what we presented in Section 3.2, we choose to set LB^{r_d} as $LB^{r_d}_{QCR^{r_d}}(W)$ or $LB^{r_d}_{UQCR^{r_d}}(W)$, defined as in Definition 3.5. In case no local upper bound belongs to LB^{r_d}, the node and all its children can be pruned. The proof follows as in [11, Lemma 4.4]:

Lemma 3.10. *Let N^{r_d} be a node and assume that problem* (MO-QCR^{r_d}) *is feasible. If the following condition holds*

$$\forall\, u \in \mathcal{U}: \;\; u \notin LB^{r_d} \tag{12}$$

then $\mathcal{N} \cap LB^{r_d} = \emptyset$, being $\mathcal{N}$ the nondominated set of (MO-BQP).

Proof. Since $W \subseteq \mathbb{R}^p_+$, for any $y \in -\mathbb{R}^p_+$ it holds that $w^\top y \leq 0$ for all $w \in W$. As a result, we have that $u \notin LB^{r_d}$ if and only if $(\{u\} - \mathbb{R}^p_+) \cap LB^{r_d} = \emptyset$. Hence, (12) holds if and only if $(\mathcal{U} - \mathbb{R}^p_+) \cap LB^{r_d} = \emptyset$. Together with (5), we obtain that if (12) holds then also $\mathcal{N} \cap LB^r = \emptyset$. $\square$

Given a node N^{r_d}, we define the boolean function `Prune(`N^{r_d}`)` as

$$\texttt{Prune(}N^{r_d}\texttt{)} := (\mathcal{F}^{r_d} = \emptyset) \vee (\forall\ u \in \mathcal{U} :\ u \notin LB^{r_d}).$$

Note that `Prune(`N^{r_d}`)` is evaluated at Step 9 of Algorithm 1 and in case (`Prune(`N^{r_d}`)` $=$ 1) node N^{r_d} is pruned. Lemma 3.9 and Lemma 3.10 imply the following:

Proposition 3.11. *If* `Prune(`N^{r_d}`)` $= 1$ *then no efficient point* $\bar{x} \in \{0,1\}^n$ *of* (MO-BQP) *is such that* $(\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_d)$.

If, at a given node N^{r_d}, the subproblems ($\mathrm{P}^{rd}(w)$) (or ($\mathrm{P}_u^{rd}(w)$)) share the same optimal solution for every $w \in W$, and this solution is binary, then we can prune the node by integrality, as shown in the next lemma. The result is stated with respect to subproblem ($\mathrm{P}^{rd}(w)$); a similar statement and proof hold when considering subproblem ($\mathrm{P}_u^{rd}(w)$).

Lemma 3.12. *Let* N^{r_d} *be a node and assume that Problem* ($\mathrm{P}^{rd}(w)$) *has the unique optimal solution* $\bar{x}^w \in \{0,1\}^{n-d}$*, for every* $w \in W$*. Let* $\bar{x} = (r_1, \ldots, r_d, \bar{x}^w) \in \{0,1\}^n$*. Then, there is no point* $\tilde{x} \in \{0,1\}^n$ *of* (MO-BQP) *with* $(\tilde{x}_1, \ldots, \tilde{x}_d) = (r_1, \ldots, r_d)$ *that dominates* $\bar{x}$*.*

Proof. First of all, note that $\bar{x} \in \{0,1\}^n$ is feasible for (MO-BQP) by construction. Assume by contradiction that $\tilde{x} \in \{0,1\}^n$, with $(\tilde{x}_1, \ldots, \tilde{x}_d) = (r_1, \ldots, r_d)$, exists such that

$$f(\tilde{x}) \leq f(\bar{x}) \ \text{ and } \ f_j(\tilde{x}) < f_j(\bar{x}) \ \text{ for some } j \in \{1, \ldots, p\}. \tag{13}$$

Since $(\tilde{x}_1, \ldots, \tilde{x}_d) = (\bar{x}_1, \ldots, \bar{x}_d) = (r_1, \ldots, r_d)$, we have that (13) is equivalent to

$$f^{r_d}(\tilde{x}) \leq f^{r_d}(\bar{x}) \ \text{ and } \ f_j^{r_d}(\tilde{x}) < f_j^{r_d}(\bar{x}) \ \text{ for some } j \in \{1, \ldots, p\}$$

and, from (6), it implies

$$(q_j^{r_d})_{(\delta_j^*, \beta_j^*)}(\tilde{x}) < (q_j^{r_d})_{(\delta_j^*, \beta_j^*)}(\bar{x}). \tag{14}$$

In particular, from (7), we have $e_j \in W$ and then, for $w = e_j$, (14) implies

$$w^\top (q^{r_d})_{(\delta_j^*, \beta_j^*)}(\tilde{x}) < w^\top (q^{r_d})_{(\delta_j^*, \beta_j^*)}(\bar{x})$$

which contradicts the optimality of $\bar{x}^w$ for ($\mathrm{P}^{rd}(w)$). □

Lemma 3.12 shows that, in the case that a binary feasible solution of (MO-BQP) is obtained by building the lower bound set at node N^{r_d}, this is a potential efficient point. In particular, it is the only possible efficient point obtained with the vector of fixings r_d. Therefore, given a node N^{r_d}, we define the boolean function `CheckInt(`N^{r_d}`)` as

$$\texttt{CheckInt(}N^{r_d}\texttt{)} := (\text{Assumption of Lemma 3.12}) \vee (r_d \in \{0,1\}^n).$$

Note that `CheckInt(`N^{r_d}`)` is evaluated at Step 11 of Algorithm 1. In case (`CheckInt(`N^{r_d}`)` $=$ 1) an update of the stable set S is performed according to the integer feasible point detected. In particular, if a leaf node has been reached, we have that in case $r_d \in \{0,1\}^n$ is a feasible binary assignment, $f(\bar{x}) = f(r_d)$ is a potential nondominated point. On the other hand, if the assumption of Lemma 3.12 are satisfied, we have that $\bar{x} = (r_1, \ldots, r_d, \bar{x}^w)$ is a potential efficient point. In both cases the stable set S is updated considering the feasible point in the image space $f(\bar{x})$.

Theorem 3.13. *Algorithm 1 stops after a finite number of iterations returning the set S, that is the nondominated set of* (MO-BQP)*.*

Proof. As Problem (MO-BQP) is binary, the search tree built by Algorithm 1 is finite. From Proposition 3.11, we have that a node is pruned only if it is related to a vector of fixings of the binary variables that cannot lead to an efficient binary assignment, or, in other words, cannot contribute to the nondominated set. Furthermore, any time a feasible point $\bar{x} \in \{0,1\}^n$ is explored by Algorithm 1, i.e. any time a node with `CheckInt(`N^{r_d}`)` $=$ 1 is reached by Algorithm 1, we have that $\bar{x}$ is a potential efficient point and the image $f(\bar{x}) \in \mathbb{R}^p$ is included in the set S, keeping S a stable set.

Then, along the iterations of Algorithm 1, S is a finite set of pairwise non-comparable points, images of feasible points of (MO-BQP). Once all the nodes have been visited and the list $\mathcal{L}$ is empty (Step 5 in Algorithm 1), all possible feasible binary assignments have been visited and the stable set S obtained as output of Algorithm 1 is the nondominated set of (MO-BQP). □

4 Numerical Results

We evaluate the performance of `MObBQ` on instances of the MultiObjective Max-Cut Problem (MO-MCP) and of the MultiObjective k-item Quadratic Knapsack Problem (MO-kQKP) with $p = 2, 3$. We implemented five different versions of `MObBQ`, depending on the quadratic convex reformulations that we use to compute the lower bound sets at the nodes:

- `MObBQ`$_{UQCR}$: the unconstrained quadratic convex reformulation (MO-UQCRrd) is adopted at every node,

- `MObBQ`$_{UQCR^*}$: an *improved* unconstrained quadratic convex reformulation is adopted at every node (see Section 4.1)

- `MObBQ`$_{QCR+UQCR}$: the constrained quadratic convex reformulation (MO-QCR) is adopted at the root node and the unconstrained quadratic convex reformulation (MO-UQCRrd) is adopted at every other node,

- `MObBQ`$_{QCR+UQCR^*}$: the constrained quadratic convex reformulation (MO-QCR) is adopted at the root node and the *improved* unconstrained quadratic convex reformulation is adopted at every other node (see Section 4.1),

- `MObBQ`$_{QCR^0}$ (or `MObBQ`$_{UQCR^0}$): the constrained (unconstrained) quadratic convex reformulation (MO-QCR) (or (MO-UQCR)) is adopted to reformulate (MO-BQP), so that the lower bound sets computed at each node is $LB^{r_d}_{QCR^0}(W)$ (or $LB^{r_d}_{UQCR^0}(W)$).

In each implementation of `MObBQ` we set $|W| = p + 1$, namely W is made of the p unit vectors plus the vector $(p, \dots, p)/\|p\|$.

In this work, the upper bound set is initialized as the empty set. For further study, single-objective heuristic algorithms, such as the primal rounding heuristic or other dedicated algorithms in [26], can be applied to multiobjective context.

All the algorithms considered have been implemented in Julia v.1.11.2. All experiments have been performed on a Linux machine with 4 Intel(R) Core(TM) i7-8650U CPUs and 8 GB RAM.

The execution time limit is set to half an hour for all experiments in this paper. To compute the optimal quadratic convex reformulation parameters, the CSDP solver v.6.2.0 `https://github.com/jump-dev/CSDP.jl` is invoked to solve semidefinite programming problems. At each node, the single-objective weighted sum quadratic convex continuous problems are solved by the Gurobi solver v12.0.0 (academic license). Our implementations are available at `https://github.com/Yue0925/MultiObjectiveAlgorithms.jl`.

On bi-objective instances, we compare our versions of `MObBQ` with the ϵ-constraint method (also implemented in Julia) available in `JuMP` at `https://jump.dev/JuMP.jl/stable/packages/MultiObjectiveAlgorithms`.

4.1 Static branching and pre-processing

The computation of lower bound sets from a specific quadratic convex reformulation of (MO-BQPrd) requires solving p semidefinite programming problems at each node. Since this incurs a non-negligible computational cost, our implementation of `MObBQ` employs static branching to allow for a pre-processing phase. This enables to perform heavy computations before starting the enumeration of the nodes and to significantly improve the overall performance of the branch-and-bound. In particular, the order in which binary variables are fixed is predetermined (assume w.l.o.g. the natural order from 1 to n) and we have that at a generic level $d \in [n] \cup \{0\}$ of the search tree, the variables $x_1, \ldots, x_d$ are fixed. Hence, at every node of the search tree belonging to the same level $d \in [n] \cup \{0\}$, the same set of binary variables is fixed to certain (different) values. This means that the subproblems at the nodes belonging to the same level $d \in [n] \cup \{0\}$ share the same matrices Q_j^d, $j \in [p]$.

In particular, if `MObBQ`$_{UQCR}$ or `MObBQ`$_{QCR+UQCR}$ is used, we have that nodes belonging to the same level share the same parameters $\delta_j^{u,*}$, $j \in [p]$ for the computation of $LB^{r_d} = LB^{r_d}_{UQCR^{r_d}}(W)$. Therefore, these parameters can be computed in a pre-processing phase by solving $p \times n$ semidefinite programming problems (namely, p semidefinite programming problems for each level $d = 0, \ldots, n-1$) of the following form:

$$
\begin{aligned}
(\mathrm{SDP}_{UQCR})_j^d \quad \min_{x} \quad & \sum_{i=1}^{n-d} \sum_{\ell=1}^{n-d} (Q_j^d)_{i\ell} \, X_{il} \\
\text{s.t.} \quad & X_{ii} = x_i, \quad i \in [n-d] \\
& \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0 \\
& x \in \mathbb{R}^{n-d}, \ X \in \mathcal{S}^{n-d},
\end{aligned}
$$

where $d = 1, \ldots, n-1$ and $j \in [p]$. We remark that at level $d = 0$ (root node) no variable is fixed so that $Q_j^d = Q_j$, $j \in [p]$ is the original j-th matrix.

For the computation $LB^{r_d} = LB^{r_d}_{QCR^{r_d}}(W)$, we cannot adopt a similar strategy, as the SDP subproblems depend on the vector of fixings r_d, except, of course, for the SDP subproblem associated with the root node. Indeed, at a node N^{r_d} with $r_d \neq \emptyset$, the constraint (17) within (SDP$_{QCR}$) (see Appendix) depends on b^{r_d}. Preliminary tests indicated that solving p semidefinite programming problems at each node proved to be too time-consuming and did not justify the computational effort, despite the high quality of the lower bound sets obtained (see Figure 1).

However, as long as the matrix A and the vector b in Problem (MO-BQP) have nonnegative entries (like, for example, for (MO-kQKP)), we can still compute in a pre-processing phase an *improved* unconstrained quadratic convex reformulation of (MO-BQPrd). Indeed, under the assumption that A and b have nonnegative entries, it holds

$$\{x \in [0,1]^{n-d} \mid A^d x \leq b^{r_d}\} \subseteq \{x \in [0,1]^{n-d} \mid A^d x \leq b\},$$

and, in particular,

$$\{x \in [0,1]^{n-d} \mid A^d_{=} x = b^{r_d}_{=}\} \subseteq \{x \in [0,1]^{n-d} \mid A^d_{=} x \leq b_{=}\}.$$

Therefore, in the pre-processing phase of $\texttt{MObBQ}_{UQCR^*}$ and $\texttt{MObBQ}_{QCR+UQCR^*}$, we solve the following semidefinite programming problem:

$$\begin{aligned} (\text{SDP}_{UQCR^*})^d_j \quad \min_x \quad & \sum_{i=1}^{n-d}\sum_{\ell=1}^{n-d} (Q_j)^d_{i\ell}\, X_{ij} \\ \text{s.t.} \quad & X_{ii} = x_i, \quad i \in [n-d] \qquad (15) \\ & A^d x \leq b \\ & \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0 \\ & x \in \mathbb{R}^{n-d},\ X \in \mathcal{S}^{n-d}, \end{aligned}$$

for every level $d = 1, \ldots, n-1$ and every $j \in [p]$. The optimal values of the dual variables associated with constraints (15) give the parameters $\tilde{\delta}^{u,*}_j \in \mathbb{R}^{n-d}$, $j \in [p]$ so that

$$\min\{(q^{r_d}_j)_\delta(x) \mid A^d x \leq b,\ x \in [0,1]^{n-d}\}$$

is maximized. These values are the ones we adopt in the implementation of $\texttt{MObBQ}_{UQCR^*}$ and $\texttt{MObBQ}_{QCR+UQCR^*}$ in order to compute the lower bound set LB^{r_d}.

4.2 Results on multiobjective max-cut instances

As a first case study, we focus on the MultiObjective Max-Cut Problem (MO-MCP). Let $G = (V, E)$ be a graph with n vertices and m edges. We consider p functions defined over the set of edges of the graph. As objective functions, we can consider e.g. the weight of an edge, the cost or the reward associated to an edge, etc...

Given a partition $\{V_1, V_2\}$ of V, a cut is defined as the set $E(V_1, V_2)$ of edges having exactly one endpoint in V_1. The multiobjective max-cut problem aims to find cuts that simultaneously maximize the p objective functions considered. As for the single objective max-cut problem (see, e.g., [2, 6]), the multiobjective max-cut problem can be equivalently formulated as a multiobjective unconstrained binary quadratic problem:

$$\begin{aligned} \max_x \quad & (f_1(x), \ldots, f_p(x))^\top \\ \text{s.t.} \quad & x_i \in \{0,1\} \qquad i \in \{1, \ldots, n\}, \end{aligned} \qquad \text{(MO-MCP)}$$

where

$$f_j(x) = \sum_{u\in V}\sum_{v\in V}(Q_j)_{uv}\big(x_u(1-x_v)+x_v(1-x_u)\big),$$

with $Q_j \in \mathcal{S}^n$, $j \in [p]$.

Problem (MO-MCP) can be addressed by MObBQ in its versions using the unconstrained quadratic reformulations. In Table 1 and Table 2 we report the results on bi- and tri-objective MO-MCP instances, respectively. The number of vertices of the graph n varies from 10 to 30 and the matrices Q_j are randomly generated with values in $[5, 50]$, with density 25%, 50%, 75% and 100%. In order to build multiple conflicting objective functions, we keep the corelation $\rho\big(f_1(x), f_2(x)\big) \approx -0.92$ for bi-objective instances and $\rho\big(f_1(x)+f_2(x), f_3(x)\big) \approx -0.92$ for tri-objective instances. In each table, we report the average number of nondominated points of the instances (rounded down) and - for each solver - the number of instances successfully solved within the time limit of 1800 seconds (solved), the average CPU time (in seconds) and the average number of nodes (rounded down). All averages are taken over the number of instances successfully solved within the time limit among the three instances built for each fixed n and density. Looking at the results in both tables, it is clear that $\texttt{MObBQ}_{UQCR}$ consistently outperforms $\texttt{MObBQ}_{UQCR^0}$ both in terms of CPU time and number of nodes needed to solve the instances. This observation aligns with Proposition 3.6, which suggests that the lower bound set $LB^{r_d}_{UQCR^{r_d}}(W)$ is stronger than $LB^{r_d}_{UQCR^0}(W)$. Consequently, the likelihood of pruning nodes that cannot lead to efficient binary assignments is increased. In Table 1 we also report a comparison with the ϵ-constraint method (see e.g. [13]). We can notice that the ϵ-constraint method is faster for instances with low-density matrices. On the other hand, for instances with full-density matrices, MObBQ is always faster, and especially for instances with $n = 25$ it is even able to solve a higher number of instances within the time limit.

4.3 Results on multiobjective k-item quadratic knapsack instances

As a second case study, we consider the MultiObjective 0-1 exact k-item Quadratic Knapsack Problem (MO-kQKP). We refer to [26, 27, 7] for details on the single-objective (kQKP). The formulation of the problem is as follows. We maximize p quadratic functions subject to a capacity and a cardinality constraint. The feasible set is indeed defined by only two linear constraints (one equation and one inequality):

$$\begin{aligned} \max_x \quad & (f_1(x), \dots, f_p(x))^\top \\ \text{s.t.} \quad & \sum_{i=1}^n a_i x_i \le b \\ & \sum_{i=1}^n x_i = k \\ & x_i \in \{0,1\} \qquad\qquad i \in \{1,\dots,n\}, \end{aligned} \tag{MO-kQKP}$$

where $f_j(x) : \mathbb{R}^n \to \mathbb{R}$ is defined as $f_j(x) = \sum_{i=1}^n \sum_{\ell=1}^n (Q_j)_{i\ell} x_i x_l$. In problem (MO-kQKP), n indicates the number of items, k the number of items to be filled in the knapsack, $a_i, i \in [n]$ the weight of the i-th item, $(Q_j)_{i\ell}$ a value associated with the selection of items i and ℓ (such, e.g.,

Table 1: Comparison between the ϵ-constraint method and `MObBQ` on bi-objective Max-Cut instances.

n	**density%**	$\lfloor$**#NDP**$\rceil$	ϵ**-constraint**		`MObBQ`$_{UQCR^0}$			`MObBQ`$_{UQCR}$		
			solved	**time(s)**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**
10	25	2	3	**0.05**	3	0.24	199	3	0.36	166
	50	7	3	**0.15**	3	0.27	263	3	0.40	176
	75	8	3	**0.29**	3	0.53	590	3	0.56	389
	100	16	3	1.49	3	**0.98**	1179	3	1.03	781
15	25	3	3	**0.10**	3	1.71	1568	3	1.15	675
	50	8	3	**0.29**	3	1.55	1581	3	1.32	889
	75	14	3	**1.77**	3	4.22	3909	3	2.29	1747
	100	54	3	29.35	3	20.26	20124	3	**11.11**	9459
20	25	14	3	**0.73**	3	7.69	6389	3	4.11	2773
	50	15	3	**1.66**	3	19.38	14960	3	6.78	4605
	75	20	3	**10.68**	3	35.97	26222	3	11.01	7870
	100	72	3	438.18	3	363.64	263236	3	**105.49**	68350
25	25	13	3	**1.01**	3	50.16	35445	3	14.25	8582
	50	19	3	**3.83**	3	63.81	40101	3	15.11	8984
	75	22	3	55.75	3	160.67	96333	3	**34.73**	20567
	100	88	0	-	0	-	-	3	**1526.79**	847327
30	25	18	3	**3.19**	3	353.29	186759	3	51.72	26214
	50	26	3	**16.29**	3	484.07	244246	3	60.02	29255
	75	32	3	234.22	2	1350.45	675664	3	**172.10**	83560
	100	-	0	-	0	-	-	0	-	-

Table 2: `MObBQ` performance on tri-objective Max-Cut instances.

n	**density%**	$\lfloor$**#NDP**$\rceil$	`MObBQ`$_{UQCR^0}$			`MObBQ`$_{UQCR}$		
			solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**
10	25	73	3	1.83	1273	3	**1.23**	862
	50	103	3	2.30	1457	3	**1.48**	953
	75	103	3	2.80	1629	3	**2.04**	1275
	100	64	3	2.02	1499	3	**1.67**	1253
15	25	898	2	1623.31	37437	3	**576.34**	20894
	50	922	0	-	-	3	**984.11**	27176
	75	750	3	1494.57	42079	3	**880.20**	28209
	100	323	3	264.96	33507	3	**184.23**	27976
20	25	-	0	-	-	0	-	-
	50	-	0	-	-	0	-	-
	75	-	0	-	-	0	-	-
	100	-	0	-	-	0	-	-

the profit) and b is the capacity of the knapsack. Without loss of generality matrices Q_j, $j \in [p]$ are assumed to be symmetric.

We take the single-objective instance generator described in [7, 26] to produce random instances, with the number of items n ranging from 20 to 50. The density of the matrices Q_j is set to 25%, 50%, 75% and 100%, with coefficients drawn from the interval $[1, 100]$. The cardinality constraint parameter k is selected from the range $[2, n/4]$, while the budget b is chosen from $[50, 30k]$. The multiple conflicting objective functions are generated in a corelation $\rho\big(f_1(x), f_2(x)\big) \approx -0.92$ and $\rho\big(f_1(x) + f_2(x), f_3(x)\big) \approx -0.92$.

In Table 3 and Table 4 we report the results on bi- and tri-objective MO-kQKP instances, respectively. In each table, we report the average number of nondominated points of the instances (rounded down) and - for each solver - the number of instances successfully solved within the time limit of 1800 seconds, the average CPU time (in seconds) and the average number of nodes (rounded down). All averages are taken over the number of instances successfully solved within the time limit among the three instances built for each fixed n and density.

Looking at the results in these two tables, we can notice that all versions of `MObBQ` show similar performances and that $\texttt{MObBQ}_{UQCR}$ is in general the best with respect to the CPU time. We can also notice that both $\texttt{MObBQ}_{UQCR^*}$ and $\texttt{MObBQ}_{QCR+UQCR^*}$ often improve the performance in terms of number of nodes of $\texttt{MObBQ}_{UQCR}$ and $\texttt{MObBQ}_{QCR+UQCR}$, respectively. In Table 5 we also report a comparison between the ϵ-constraint method and $\texttt{MObBQ}_{UQCR}$ for bi-objective instances, related to the CPU time required to solve the instances. We can notice that the ϵ-constraint method is in general faster with respect to $\texttt{MObBQ}_{UQCR}$, that anyway shows similar performances for instances with n up to 35. We underline that the ϵ-constraint method is a criterion space algorithm and then has the advantage of working in a much lower dimensional space contrary to `MObBQ`.

5 Conclusions

In this paper, we introduced `MObBQ`, a branch-and-bound framework for solving MultiObjective Binary Quadratic Problems (MOBQPs). Leveraging the Quadratic Convex Reformulation (QCR) method, an MOBQP can be reformulated into a convex multiobjective binary quadratic problem. However, we demonstrate - both theoretically and numerically - that applying the QCR method at every node of the branch-and-bound tree yields stronger lower bound sets compared to a one-time reformulation. This enhanced approach incurs the cost of solving several semidefinite programming problems at each node -specifically, one for each objective function. To mitigate the related computational effort, we employ static branching, which enables a preprocessing phase and allows for faster node enumeration. We show that `MObBQ` correctly identifies the complete set of nondominated solutions for MOBQPs. Computational experiments highlight the good performance of `MObBQ`, which is the first branch-and-bound method designed to solve this class of multiobjective non-convex problems and can outperform the ϵ-constraint method on bi-objective Max-Cut instances with high-density matrices.

Table 3: MObBQ performance on bi-objective kQKP instances.

n	density%	⌊#NDP⌋	$\texttt{MObBQ}_{QCR^0}$			$\texttt{MObBQ}_{UQCR}$			$\texttt{MObBQ}_{QCR+UQCR}$			$\texttt{MObBQ}_{UQCR^*}$			$\texttt{MObBQ}_{QCR+UQCR^*}$		
			solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**
20	25	46	3	15.28	12569	3	10.58	12685	3	13.85	12692	3	**10.49**	12684	3	10.89	12691
	50	26	3	3.55	2814	3	**2.87**	2943	3	3.85	2945	3	3.42	2940	3	3.37	2942
	75	29	3	7.76	6586	3	**6.11**	6981	3	7.47	7049	3	6.66	6073	3	7.20	7044
	100	27	3	2.58	1874	3	**2.34**	1917	3	2.68	1919	3	2.54	1911	3	2.86	1911
25	25	39	3	77.11	57986	3	236.50	70366	3	81.50	70375	3	**69.24**	70320	3	70.22	70329
	50	36	3	7.82	5579	3	**6.11**	6119	3	8.90	6123	3	6.82	6118	3	7.28	6122
	75	54	3	38.02	28010	3	**30.81**	28587	3	37.15	28596	3	34.17	28356	3	34.50	28365
	100	29	3	2.91	1721	3	**2.57**	1734	3	4.17	1734	3	2.75	1725	3	3.69	1725
30	25	37	3	102.27	66127	3	**79.44**	72409	3	91.04	72409	3	82.60	72283	3	83.17	72283
	50	58	2	29.84	19899	2	**22.98**	20033	2	29.25	20033	2	27.04	19740	2	25.34	19740
	75	34	3	23.48	15398	3	**19.22**	16241	3	26.79	16640	3	21.13	15854	3	22.84	16378
	100	30	3	6.68	3484	3	**6.30**	3529	3	8.03	3529	3	6.69	3507	3	7.61	3507
35	25	31	3	48.00	27566	3	**37.07**	27769	3	44.06	27769	3	39.91	27703	3	41.81	27703
	50	71	2	88.21	52597	2	**73.51**	53943	2	88.14	53943	2	76.26	53800	2	77.17	53800
	75	55	3	238.94	140667	3	**194.27**	152734	3	234.87	152734	3	203.16	151863	3	203.50	151863
	100	31	3	8.90	3848	3	**7.88**	3881	3	13.78	3881	3	10.21	3875	3	11.48	3875
40	25	43	3	444.06	226818	3	**331.12**	235557	3	391.68	235557	3	340.79	235377	3	342.21	235377
	50	57	3	293.61	140495	3	**241.28**	155633	3	271.17	157365	3	247.49	148392	3	269.34	152065
	75	65	2	114.15	56598	2	**85.61**	57303	2	117.77	57306	2	92.68	57115	2	98.05	57118
	100	74	2	705.46	351445	2	**602.865**	393893	2	712.47	393893	2	659.09	369361	2	678.43	369361
45	25	137	1	62.30	27815	1	93.02	27821	1	**56.55**	27821	1	52.38	27821	1	56.71	27821
	50	105	1	**10.76**	1979	1	29.01	1979	1	19.4	1979	1	14.21	1979	1	20.73	1979
	75	39	2	540.60	218681	2	**415.60**	225803	2	434.05	225803	2	433.535	222418	2	442.89	222418
	100	44	2	34.68	12186	2	**31.74**	12211	2	52.37	12211	2	37.74	12175	2	43.16	12175
50	25	74	1	20.80	2355	1	**20.56**	2355	1	43.61	2355	1	24.16	2355	1	42.88	2355
	50	61	2	556.225	169760	2	**383.525**	170908	2	424.905	170908	2	393.72	169676	2	504.26	169676
	75	87	2	533.51	167775	2	**413.85**	179285	2	465.08	179543	2	440.50	171262	2	501.24	171649
	100	50	2	**41.12**	13055	2	43.44	13071	2	58.23	13071	2	53.16	13038	2	64.75	13038

Table 4: MObBQ performance on tri-objective kQKP instances.

n	density%	⌊#NDP⌋	$MObBQ_{QCR^0}$			$MObBQ_{UQCR}$			$MObBQ_{QCR+UQCR}$			$MObBQ_{UQCR^*}$			$MObBQ_{QCR+UQCR^*}$		
			solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**	solved	**time(s)**	**#nodes**
20	25	220	3	45.36	12660	3	**39.29**	12692	3	40.06	12692	3	40.27	12692	3	46.30	12692
	50	127	3	8.55	2927	3	**6.05**	2946	3	6.27	2947	3	6.39	2945	3	7.53	2947
	75	176	3	56.10	6962	3	**47.67**	7055	3	81.04	7065	3	48.32	7055	3	58.28	7065
	100	107	3	7.04	1924	3	**4.51**	1928	3	4.67	1928	3	5.09	1928	3	6.04	1928
25	25	79	2	5.22	2333	2	**4.86**	2333	2	9.98	2333	2	5.08	2333	2	6.61	2333
	50	145	3	19.76	6038	3	**15.97**	6137	3	33.44	6137	3	16.79	6137	3	18.88	6137
	75	383	3	315.41	28759	3	**285.27**	28924	3	466.93	28924	3	293.24	28907	3	323.55	28907
	100	114	3	**5.77**	1734	3	7.06	1736	3	11.76	1736	3	6.12	1736	3	7.95	1736
30	25	61	2	6.545	2006	2	**5.22**	2028	2	15.26	2028	2	6.68	2027	2	9.66	2027
	50	237	2	162.11	20637	2	**143.20**	20679	2	144.42	20686	2	147.97	20681	2	166.48	20682
	75	265	3	161.18	16847	3	**147.69**	17011	3	146.03	17128	3	148.77	16879	3	163.02	17049
	100	153	3	12.19	3533	3	**10.27**	3542	3	20.41	3542	3	13.27	3541	3	18.69	3541
35	25	171	3	270.72	27835	3	**236.37**	27888	3	239.11	27888	3	243.97	27871	3	270.13	27871
	50	436	2	501.59	55138	2	436,44	55450	2	**395.25**	55450	2	456.71	55435	2	451.45	55435
	75	555	1	1512.78	69713	1	891.07	69927	1	**854.96**	69927	1	910.57	69875	1	919.18	69875
	100	137	3	17.54	3899	3	**15.53**	3900	3	17.88	3900	3	19.53	3900	3	25.28	3900
40	25	221	2	482.09	47744	2	411.685	47851	2	**410.66**	47851	2	412.37	47802	2	430.445	47802
	50	374	2	733.46	59638	2	**624.01**	59680	2	631.51	59680	2	632.89	59680	2	698.925	59680
	75	125	1	23.67	3767	1	**17.18**	3767	1	31.13	3767	1	27.77	3767	1	58.31	3767
	100	413	3	517.62	41455	3	320.45	41467	3	**320.05**	41467	3	336.99	41461	3	377.71	41461
45	25	284	1	163.57	27821	1	137.04	27821	1	137.39	27821	1	**135.42**	27821	1	153.48	27821
	50	105	1	17.24	1979	1	**24.17**	1979	1	34.89	1979	1	30.25	1979	1	51.38	1979
	75	23	1	**21.23**	897	1	21.35	917	1	36.50	917	1	34.71	917	1	54.62	917
	100	183	2	79.53	12235	2	**69.86**	12237	2	80.71	12237	2	84.18	12236	2	123.63	12236
50	25	57	1	**31.53**	2355	1	33.26	2355	1	59.79	2355	1	49.19	2355	1	89.58	2355
	50	452	1	874.31	76037	1	**731.13**	76027	1	747.94	76027	1	792.64	75749	1	795.04	75749
	75	473	1	1412.51	130025	1	**1232.07**	136253	1	1296.24	136293	1	1399.77	134413	1	1514.37	134495
	100	282	2	**126.66**	13080	2	127.15	13083	2	140.105	13083	2	140.725	13082	1	169.47	13082

Table 5: Comparison with the ϵ-constraint method with `MObBQ` on bi-objective kQKP instances.

n	**density%**	$\lfloor$**#NDP**$\rfloor$	ϵ-constraint		$\texttt{MObBQ}_{UQCR}$	
			solved	**time(s)**	solved	**time(s)**
20	25	46	3	**7.40**	3	10.58
	50	26	3	3.69	3	**2.87**
	75	29	3	**3.94**	3	6.11
	100	27	3	6.26	3	**2.34**
25	25	39	3	**11.09**	3	236.50
	50	36	3	7.01	3	**6.11**
	75	54	3	**21.32**	3	30.81
	100	29	3	6.26	3	**2.57**
30	25	37	3	**13.03**	3	79.44
	50	58	**3**	47.89	2	22.98
	75	34	3	**19.03**	3	19.22
	100	30	3	8.26	3	**6.30**
35	25	31	3	**19.46**	3	37.07
	50	71	**3**	76.61	2	73.51
	75	55	3	**47.67**	3	194.27
	100	31	3	12.94	3	**7.88**
40	25	43	3	**45.82**	3	331.12
	50	57	3	**82.65**	3	241.28
	75	65	**3**	110.90	2	85.61
	100	74	**3**	217.96	2	602.865
45	25	137	**3**	649.99	1	93.02
	50	105	**3**	878.81	1	29.01
	75	39	2	**118.28**	2	415.60
	100	44	**3**	77.30	2	31.74
50	25	74	1	**6.83**	1	20.56
	50	61	**3**	126.37	2	383.525
	75	87	**3**	664.71	2	413.85
	100	50	**3**	201.87	2	43.44

Statements and Declarations

Funding

The work has been partially undertook while Marianna De Santis was visiting LIPN - Université Sorbonne Paris Nord.

Competing interests

The authors declare they have no financial interests.

Data availability statement

The data presented in this manuscript are reproducible through the implementation publicly available on `https://github.com/Yue0925/MultiObjectiveAlgorithms.jl`.

Declaration of generative AI and AI-assisted technologies in the writing process.

During the preparation of this work the authors used ChatGPT solely to refine the grammar and enhance the readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix

From [5], the optimal value of Problem (1) is equal to the optimal value of the following semidefinite programming problem

$$
\begin{aligned}
(\mathrm{SDP}_{QCR}) \min_{x} \quad & c^\top x + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}X_{ij} \\
\text{s.t.} \quad & X_{ii} = x_i, \quad i \in [n] && (16)\\
& \langle A_= A_=^\top, X\rangle - 2b_=^\top A_= x = -b_=^2 && (17)\\
& Ax \le b \\
& \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0 \\
& x \in \mathbb{R}^n,\ X \in \mathcal{S}^n,
\end{aligned}
$$

where with $\langle X, Y\rangle = trace(XY)$ we denote the standard inner product in the set $\mathcal{S}^n$ of n-by-n symmetric matrices. The parameters $(\delta^*, \beta^*) \in \mathbb{R}^{n+1}$ employed in the quadratic convex reformulation $(\mathrm{QCR}_{(\delta,\beta)})$ defined in Section 1 are given by the optimal values of the dual variables associated with constraints (16) and (17), respectively. The semidefinite programming problem addressed to define the quadratic convex reformulation (UQCR_δ) (see again Section 1) is the following

$$
\begin{aligned}
(\text{SDP}_{UQCR}) \ \min_{x} \quad & c^\top x + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}X_{ij} \\
\text{s.t.} \quad & X_{ii} = x_i, \quad i \in [n] \qquad (18)\\
& \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0 \\
& x \in \mathbb{R}^n, \ X \in \mathcal{S}^n.
\end{aligned}
$$

The parameters $\delta^{u,*} \in \mathbb{R}^n$ are obtained as the optimal values of the dual variables associated with constraints (18).

References

[1] Lavinia Amorosi, Luca Cedola, Paolo Dell'Olmo, and Francesca Lucchetta. Multi-objective mathematical programming for optimally sizing and managing battery energy storage for solar photovoltaic system integration of a multi-apartment building. *Engineering Optimization*, 54(1):81–100, 2022.

[2] Francisco Barahona, Michael Jünger, and Gerhard Reinelt. Experiments in quadratic 0–1 programming. *Mathematical programming*, 44(1):127–137, 1989.

[3] Alain Billionnet and Sourour Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Math. Program.*, 109:55–68, 2007.

[4] Alain Billionnet, Sourour Elloumi, and Amélie Lambert. Extending the qcr method to general mixed-integer programs. *Math. Program.*, 131(1):381–401, 2012.

[5] Alain Billionnet, Sourour Elloumi, and Marie-Christine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discret. Appl. Math.*, 157(6):1185–1197, 2009.

[6] Endre Boros and Peter L Hammer. The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, 33(3):151–180, 1991.

[7] Alberto Ceselli, Lucas Létocart, and Emiliano Traversi. Dantzig–wolfe reformulations for binary quadratic problems. *Math. Program. Comput.*, 14(1):85–120, 2022.

[8] Vira Chankong and Yacov Y Haimes. Multiobjective decision making theory and methodology. *Elsevier Science Publishing, New York*, 1983.

[9] Marianna De Santis and Gabriele Eichfelder. A decision space algorithm for multiobjective convex quadratic integer optimization. *Comput. Oper. Res.*, 134:105396, 2021.

[10] Marianna De Santis, Gabriele Eichfelder, Julia Niebling, and Stefan Rocktäschel. Solving multiobjective mixed integer convex optimization problems. *SIAM J. Optim.*, 30(4):3122–3145, 2020.

[11] Marianna De Santis, Gabriele Eichfelder, Daniele Patria, and Leo Warnow. Using dual relaxations in multiobjective mixed-integer convex quadratic programming. *J. Global Optim.*, 2024.

[12] Marianna De Santis, Giorgio Grani, and Laura Palagi. Branching with hyperplanes in the criterion space: The frontier partitioner algorithm for biobjective integer programming. *European J. Oper. Res.*, 283(1):57–69, 2020.

[13] Matthias Ehrgott. *Multicriteria Optimization.* Springer Science & Business Media, Heidelberg, 2005.

[14] Matthias Ehrgott and Xavier Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Comput. Oper. Res.*, 34(9):2674–2694, 2007.

[15] Gabriele Eichfelder, Peter Kirst, Laura Meng, and Oliver Stein. A general branch-and-bound framework for continuous global multiobjective optimization. *J. Global Optim.*, 80(1):195–227, 2021.

[16] Gabriele Eichfelder, Moritz Link, Stefan Volkwein, and Leo Warnow. An adaptive relaxation-refinement scheme for multi-objective mixed-integer nonconvex optimization. Preprint 27337, Optimization Online, 2024.

[17] Gabriele Eichfelder, Oliver Stein, and Leo Warnow. A solver for multiobjective mixed-integer convex and nonconvex optimization. *J. Optim. Theory Appl.*, 203(2):1736–1766, 2024.

[18] Gabriele Eichfelder and Leo Warnow. An approximation algorithm for multi-objective optimization problems using a box-coverage. *J. Global Optim.*, 83(2):329–357, 2022.

[19] Gabriele Eichfelder and Leo Warnow. Advancements in the computation of enclosures for multi-objective optimization problems. *European J. Oper. Res.*, 310(1):315–327, 2023.

[20] Gabriele Eichfelder and Leo Warnow. A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. *Math. Methods Oper. Res.*, 2023. DOI: 10.1007/s00186-023-00828-x.

[21] Nicolas Forget, Sune Lauth Gadegaard, Kathrin Klamroth, Lars Relund Nielsen, and Anthony Przybylski. Branch-and-bound and objective branching with three or more objectives. *Comput. Oper. Res.*, 148:106012, 2022.

[22] Pascal Halffmann, Luca E. Schäfer, Kerstin Dächert, Kathrin Klamroth, and Stefan Ruzika. 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.

[23] Pubudu L.W. Jayasekara Merenchige and Margaret Wiecek. A branch and bound algorithm for biobjective mixed integer quadratic programs. Preprint 21294, Optimization Online, 2022.

[24] Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. On the representation of the search region in multi-objective optimization. *European J. Oper. Res.*, 245(3):767–778, 2015.

[25] Audrey Legendre, Eric Angel, and Fariza Tahi. Bi-objective integer programming for RNA secondary structure prediction with pseudoknots. *BMC Bioinformatics*, 19(1):13, 2018.

[26] Lucas Létocart, Marie-Christine Plateau, and Gérard Plateau. An efficient hybrid heuristic method for the 0-1 exact k-item quadratic knapsack problem. *Pesquisa Operacional*, 34:49–72, 2014.

[27] Lucas Létocart and Angelika Wiegele. Exact solution methods for the k-item quadratic knapsack problem. In *Combinatorial Optimization: 4th International Symposium, ISCO 2016, Vietri sul Mare, Italy, May 16-18, 2016, Revised Selected Papers 4*, pages 166–176. Springer, 2016.

[28] Moritz Link and Stefan Volkwein. Adaptive piecewise linear relaxations for enclosure computations for nonconvex multiobjective mixed-integer quadratically constrained programs. *Journal of Global Optimization*, 87(1):97–132, 2023.

[29] George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.

[30] Andrew C Pangia and Margaret M Wiecek. A branch-and-bound algorithm for parametric mixed-binary nonlinear programs. *Journal of Global Optimization*, pages 1–25, 2024.

[31] Sophie N. Parragh and Fabien Tricoire. Branch-and-bound for bi-objective integer programming. *INFORMS J. Comput.*, 31(4):805–822, 2019.

[32] Filippo Pecci and Ivan Stoianov. Bounds and convex heuristics for bi-objective optimal experiment design in water networks. *Comput. Oper. Res.*, 153:106181, 2023.

[33] Anthony Przybylski and Xavier Gandibleux. Multi-objective branch and bound. *European J. Oper. Res.*, 260(3):856–872, 2017.

[34] Aly-Joy Ulusoy, Filippo Pecci, and Ivan Stoianov. Bi-objective design-for-control of water distribution networks with global bounds. *Optim. Eng.*, 23(1):527–577, 2022.

[35] Han Zhong, Wei Guan, Wenyi Zhang, Shixiong Jiang, and Lingling Fan. A bi-objective integer programming model for partly-restricted flight departure scheduling. *PLOS ONE*, 13(5):e0196146, 2018.