Exact approaches for multi-objective binary quadratic optimization

Marianna De Santis

Università degli studi di Firenze

Global Optimization Laboratory (GOL)

PGMODAYS 2025

Outline of the talk

Outline of the talk

- Basic Definitions
- Challenges in solving MOMINLPs
- Approximating the nondominated set of a MOMINLP: the *enclosure* concept
- Branch-and-bound methods for MOMINLPs:
  - Computation of upper bound sets
  - Computation of lower bound sets
  - Pruning conditions

Outline of the talk

- Basic Definitions
- Challenges in solving MOMINLPs
- Approximating the nondominated set of a MOMINLP: the *enclosure* concept
- Branch-and-bound methods for MOMINLPs:
    - Computation of upper bound sets
    - Computation of lower bound sets
    - Pruning conditions
- A Branch-and-Bound method for Multiobjective Binary Quadratic Problems (MO-BQPs)
    - Extending the QCR paradigm to the MO setting

Multiobjective MINLPs

A **Multiobjective Mixed Integer Nonlinear** programming problem (MOMINLP) can be formulated as follows:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & (f_1(x), \dots, f_p(x))^T \\ \text{s.t.} \quad & g_k(x) \leq 0 \;\; k = 1, \dots, m \qquad\qquad \text{(MOMINLP)} \\ & x_i \in \mathbb{Z} \;\; \forall i \in I, \end{aligned}$$

Multiobjective MINLPs

A **Multiobjective Mixed Integer Nonlinear** programming problem (MOMINLP) can be formulated as follows:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & (f_1(x), \dots, f_p(x))^T \\ \text{s.t.} \quad & g_k(x) \leq 0 \;\; k = 1, \dots, m \qquad \text{(MOMINLP)} \\ & x_i \in \mathbb{Z} \;\; \forall i \in I, \end{aligned}$$

where

- $f_j, g_k : \mathbb{R}^n \to \mathbb{R};\; j = 1, \dots, p;\; k = 1, \dots, m$
- the index set $I \subseteq \{1, \dots, n\}$ specifies which variables have to take integer values

MOMINLP: a powerful modeling tool!

Multiobjective mixed integer optimization problems arise in **many application fields** such as

- engineering
- finance
- design of water/gas distribution networks
- location or production planning
- emergency management

see e.g.
[Benvenuti et al. MMOR (2025)], [Pecci and Stoianov C&OR (2023)], [Amorosi et al. EngOpt (2022)], [Liu et al. C&OR (2014)], [Yenisey et al. Omega (2014)], [Ehrgott et al. INFOR (2009)],...

Basic definitions in multiobjective optimization

- point $x^* \in \mathcal{F}$ is **efficient** for (MOMINLP) if there is no $x \in \mathcal{F}$ with $f(x) \le f(x^*)$ and $f(x) \neq f(x^*)$

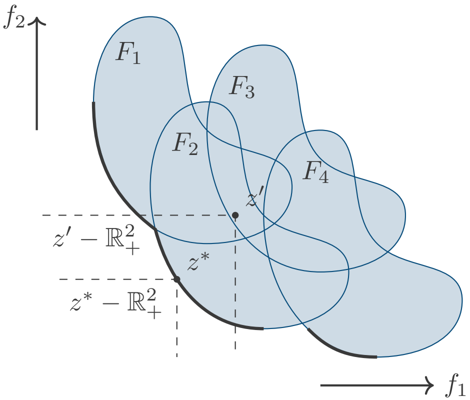
  The set of efficient points for (MOMINLP) is the **efficient set** of (MOMINLP)

- point $z^* = f(x^*) \in \mathbb{R}^p$ is **nondominated** for (MOMINLP) if $x^* \in \mathcal{F}$ is an efficient point for (MOMINLP)

  The set of all nondominated points of (MOMINLP) is the **nondominated set** of (MOMINLP)

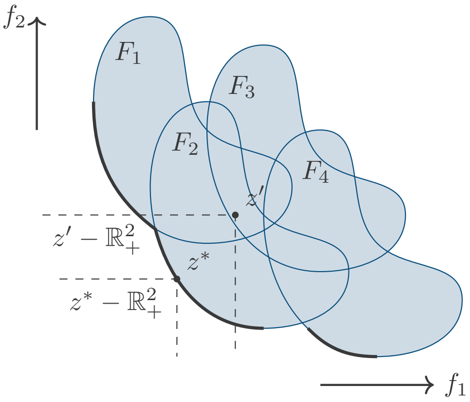
Challenges in solving MOMINLPs

Example: feasible set in the image space of a bi-objective minlp instance



Challenges in solving MOMINLPs

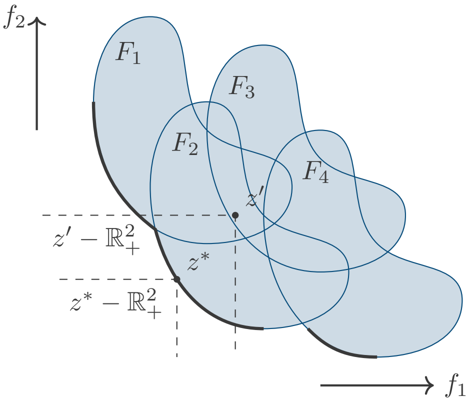
Example: feasible set in the image space of a bi-objective minlp instance



- the union of all F_j describes the whole feasible set in the image space

Challenges in solving MOMINLPs

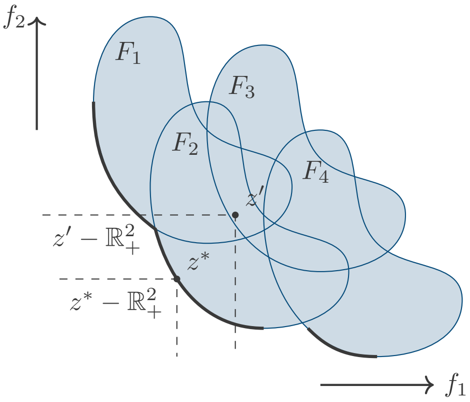
Example: feasible set in the image space of a bi-objective minlp instance



- the union of all F_j describes the whole feasible set in the image space
- z^* is a nondominated point and the preimage of z^* is an efficient point

Challenges in solving MOMINLPs

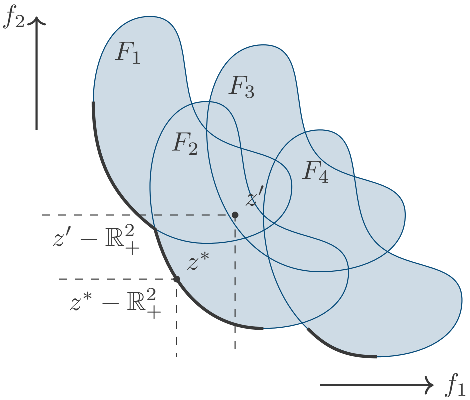
Example: feasible set in the image space of a bi-objective minlp instance



- the union of all F_j describes the whole feasible set in the image space
- z^* is a nondominated point and the preimage of z^* is an efficient point
- z' is dominated because $z^* \leq z'$ and $z^* \neq z'$.

Challenges in solving MOMINLPs

Example: feasible set in the image space of a bi-objective minlp instance

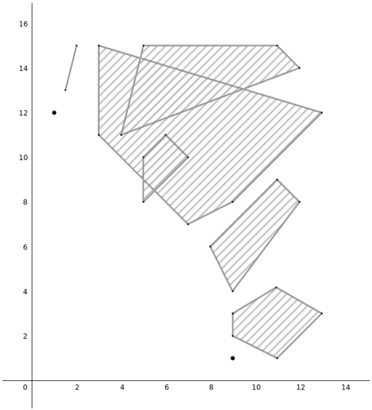


- the union of all F_j describes the whole feasible set in the image space
- z^* is a nondominated point and the preimage of z^* is an efficient point
- z' is dominated because $z^* \leq z'$ and $z^* \neq z'$.
- all the points $z \in F_3$ are dominated

...little digression on BOMILPs!

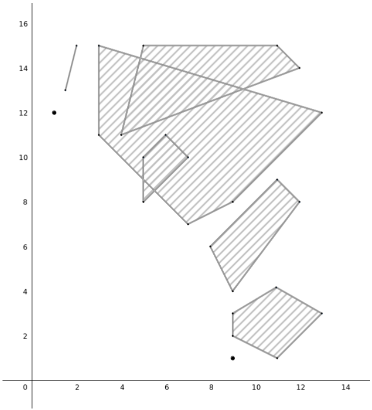
Bi-objective Mixed Integer Linear Programs

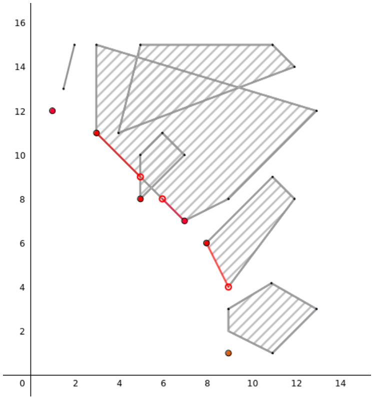
Example from [Fattahi and Turkay. *A one direction search method to find the exact nondominated frontier of biobjective mixed-binary linear programming problems.* EJOR, 2018.]



Bi-objective Mixed Integer Linear Programs

Example from [Fattahi and Turkay. *A one direction search method to find the exact nondominated frontier of biobjective mixed-binary linear programming problems.* EJOR, 2018.]





Solution approaches for MOMINLPs

Algorithms for multi-objective **mixed-integer** optimization problems compute an **approximation of the efficient or nondominated set**

Solution approaches for MOMINLPs

Algorithms for multi-objective **mixed-integer** optimization problems compute an **approximation of the efficient or nondominated set**

These approximation algorithms can be divided into two categories:

- algorithms that compute a **representation**, i.e., a finite subset, of the efficient or nondominated set

Solution approaches for MOMINLPs

Algorithms for multi-objective **mixed-integer** optimization problems compute an **approximation of the efficient or nondominated set**

These approximation algorithms can be divided into two categories:

- algorithms that compute a **representation**, i.e., a finite subset, of the efficient or nondominated set
- algorithms that compute a **coverage**, i.e., a superset, of the efficient or nondominated set

Approximating the nondominated set

In global approaches for single objective optimization the optimal solution $y^* = f(x^*) \in \mathbb{R}$ is approximated by some **lower bound** $l \in \mathbb{R}$ and some **upper bound** $u \in \mathbb{R}$ so that $l \leq y^* \leq u$,

Approximating the nondominated set

In global approaches for single objective optimization the optimal solution $y^* = f(x^*) \in \mathbb{R}$ is approximated by some **lower bound** $l \in \mathbb{R}$ and some **upper bound** $u \in \mathbb{R}$ so that $l \leq y^* \leq u$, or, equivalently,

$$\{y^*\} \subseteq (\{l\} + \mathbb{R}_+) \cap (\{u\} - \mathbb{R}_+)$$

Approximating the nondominated set

In global approaches for single objective optimization the optimal solution $y^* = f(x^*) \in \mathbb{R}$ is approximated by some **lower bound** $l \in \mathbb{R}$ and some **upper bound** $u \in \mathbb{R}$ so that $l \le y^* \le u$, or, equivalently,

$$\{y^*\} \subseteq (\{l\} + \mathbb{R}_+) \cap (\{u\} - \mathbb{R}_+)$$

This idea has been generalized to the multiobjective setting as:

$$\mathcal{N} \subseteq (L + \mathbb{R}^p_+) \cap (U - \mathbb{R}^p_+),$$

where $L, U \subseteq \mathbb{R}^p$ are nonempty sets.

[G. Eichfelder, P. Kirst, L. Meng, and O. Stein. "A general branch-and-bound framework for continuous global multiobjective optimization". JOGO 80(1) (2021), 195-227]

Definition of Enclosure

From [G. Eichfelder and L. Warnow. “An approximation algorithm for multi-objective optimization problems using a box-coverage”. JOGO 83(2) (2021), 329-357]:

Definition

Let $L, U \subseteq \mathbb{R}^p$ be two nonempty, finite sets and $N \subseteq \mathbb{R}^p$ a nonempty set.

Definition of Enclosure

From [G. Eichfelder and L. Warnow. “An approximation algorithm for multi-objective optimization problems using a box-coverage”. JOGO 83(2) (2021), 329-357]:

Definition

Let $L, U \subseteq \mathbb{R}^p$ be two nonempty, finite sets and $N \subseteq \mathbb{R}^p$ a nonempty set. Then the set

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

is called (box) coverage given L and U.

Definition of Enclosure

From [G. Eichfelder and L. Warnow. "An approximation algorithm for multi-objective optimization problems using a box-coverage". JOGO 83(2) (2021), 329-357]:

Definition

Let $L, U \subseteq \mathbb{R}^p$ be two nonempty, finite sets and $N \subseteq \mathbb{R}^p$ a nonempty set. Then the set

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

is called (box) coverage given L and U.

If $N \subseteq \mathcal{C}$, we call $\mathcal{C} \subseteq \mathbb{R}^p$ an **enclosure** of $N \subseteq \mathbb{R}^p$.

Example of enclosure $\mathcal{C}(L, U)$

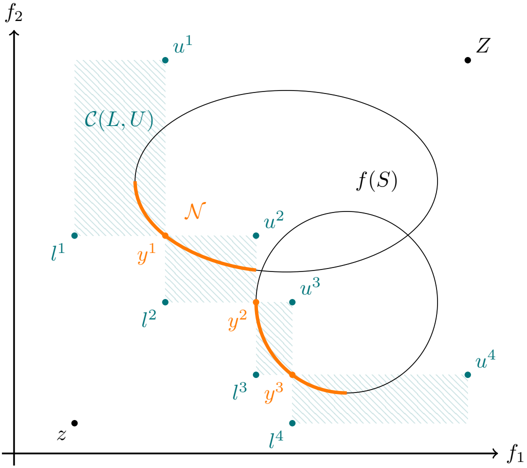


Figure: Enclosure $\mathcal{C}(L, U) \subseteq \mathbb{R}^2$ of $N = \{y^1, y^2, y^3\}$, with $L = \{l^1, l^2, l^3, l^4\}$ and $U = \{u^1, u^2, u^3, u^4\}$.
(Thanks to Dr. Leo Warnow (TU Ilmenau) for the picture!)

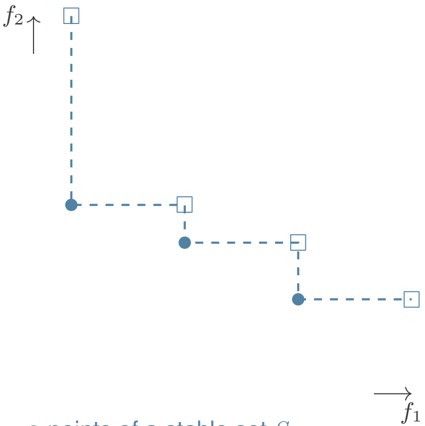
Branch-and-Bound methods for MOMINLPs

main ingredients

- Computation of upper bound sets
- Computation of lower bound sets
- Pruning rules
- Branching rules

MO Branch-and-Bound ingredients

Computation of upper bound sets



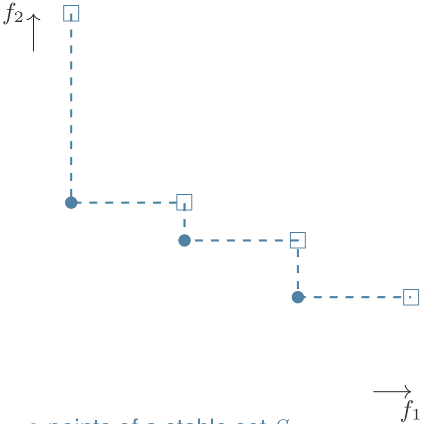
Let $S \subset \mathbb{R}^p$ be a finite stable set of images of feasible points of (MO-BQP).

● points of a stable set S

□ local upper bounds defining the upper bound set $\mathcal{U} = U(S)$

MO Branch-and-Bound ingredients

Computation of upper bound sets



Let $S \subset \mathbb{R}^p$ be a finite stable set of images of feasible points of (MO-BQP).

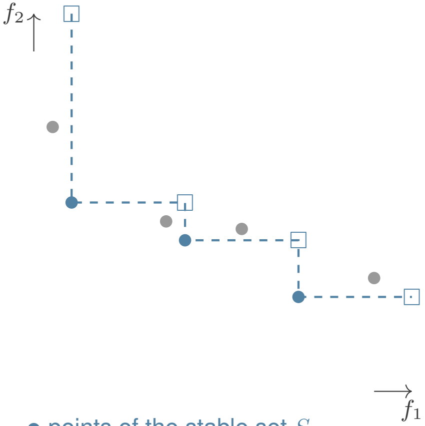
An upper bound set $\mathcal{U} = U(S)$ is obtained computing the local upper bounds using the algorithm proposed in [K. Klamroth et al. "On the representation of the search region in multi-objective optimization". EJOR 245(3), (2015), 767-778]

● points of a stable set S

□ local upper bounds defining the upper bound set $\mathcal{U} = U(S)$

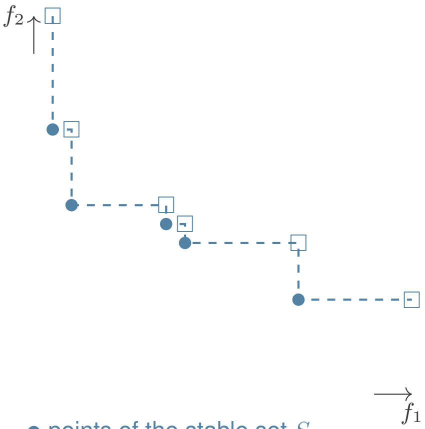
MO Branch-and-Bound ingredients

Computation of upper bound sets



As soon as new feasible points are found we try to include their images within the stable set S to improve the upper bound set $\mathcal{U}$

- ● points of the stable set S
- □ local upper bounds defining the upper bound set $\mathcal{U} = U(S)$
- ● images of new feasible points found



- S is kept as a stable set
- the upper bound set $\mathcal{U} = U(S)$ is updated with the algorithm in [K. Klamroth et al. "On the representation of the search region in multi-objective optimization". EJOR 245(3), (2015), 767-778]
- $\mathcal{N} \subseteq \mathcal{U} - \mathbb{R}^p_+$,

● points of the stable set S

□ updated local upper bounds defining the upper bound set $\mathcal{U} = U(S)$

Given (MOMINLP), the **upper image set** of its continuous relaxation is defined as

$$\mathcal{P} := \{f(x) \in \mathbb{R}^p \mid g(x) \leq 0 \; x \in \mathbb{R}^n\} + \mathbb{R}^p_+$$

Given (MOMINLP), the **upper image set** of its continuous relaxation is defined as

$$\mathcal{P} := \{f(x) \in \mathbb{R}^p \mid g(x) \leq 0 \; x \in \mathbb{R}^n\} + \mathbb{R}^p_+$$

We say that $LB \subseteq \mathbb{R}^p$ is a **lower bound set** for (MOMINLP) if

$$\mathcal{P} \subseteq LB + \mathbb{R}^p_+$$

Computation of lower bound sets

Linear supporting hyperplanes [M.Ehrgott and X. Gandibleux. “Bound sets for biobjective combinatorial optimization problems.” C&OR 34(9) (2007), 2674-2694]

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

Computation of lower bound sets

Linear supporting hyperplanes [M.Ehrgott and X. Gandibleux. “Bound sets for biobjective combinatorial optimization problems.” C&OR 34(9) (2007), 2674-2694]

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image set $\mathcal{P}$ are **obtained by solving single-objective subproblems with** $w \in W$:

Computation of lower bound sets

Linear supporting hyperplanes [M.Ehrgott and X. Gandibleux. “Bound sets for biobjective combinatorial optimization problems.” C&OR 34(9) (2007), 2674-2694]

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image set $\mathcal{P}$ are **obtained by solving single-objective subproblems with $w \in W$**:

$$\theta(w) := \min\{w^\top f(x) \mid g(x) \leq 0, x \in \mathbb{R}^n\}$$

Computation of lower bound sets

Linear supporting hyperplanes [M.Ehrgott and X. Gandibleux. “Bound sets for biobjective combinatorial optimization problems.” C&OR 34(9) (2007), 2674-2694]

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image set $\mathcal{P}$ are **obtained by solving single-objective subproblems with $w \in W$**:

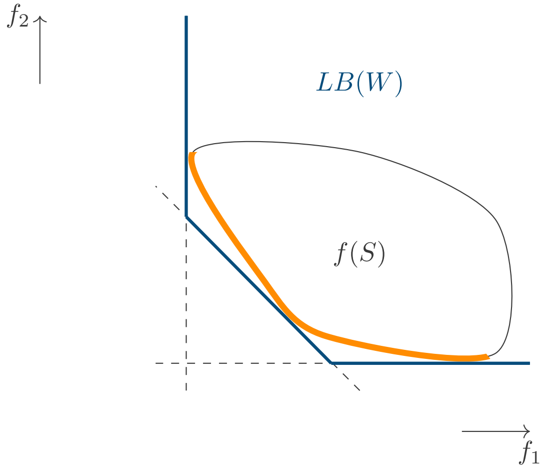
$$\theta(w) := \min\{w^\top f(x) \mid g(x) \leq 0, x \in \mathbb{R}^n\}$$

A **lower bound set** can then be defined as

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

Computation of lower bound sets

$W = \{(1,0),\ (0,1),\ (0.5,0.5)\}$



$S = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$

MO Branch-and-Bound ingredients

Pruning condition: comparing $\mathcal{U}$ with LB

$$\forall\, u \in \mathcal{U} :\;\; u \notin LB + \mathbb{R}^p_+. \qquad \text{(Cond)}$$

MO Branch-and-Bound ingredients

Pruning condition: comparing $\mathcal{U}$ with LB

$$\forall\, u \in \mathcal{U} :\; u \notin LB + \mathbb{R}^p_+. \qquad \text{(Cond)}$$

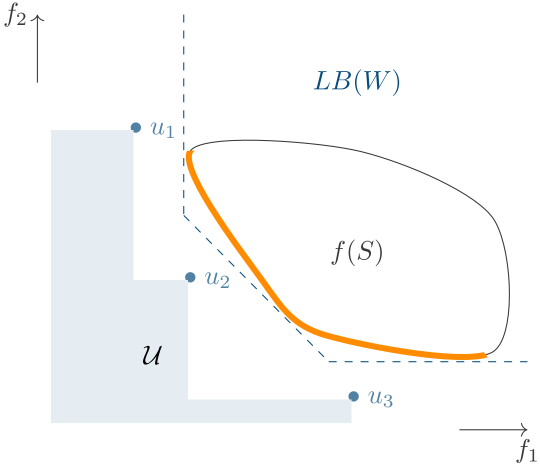
Lemma

Assume that $S \neq \emptyset$*. If* (Cond) *holds then for the nondominated set* $\mathcal{N}$ *of (MOMINLP) we have* $\mathcal{N} \cap (LB + \mathbb{R}^p_+) = \emptyset$

MO Branch-and-Bound ingredients

Pruning condition: comparing $\mathcal{U}$ with LB

$S \neq \emptyset$



MO Branch-and-Bound ingredients

Branching rules

- ※ Spatial branch-and-bound methods:

intervals are partitioned for each variable

[G. Eichfelder, O. Stein, L. Warnow; "A solver for multiobjective mixed-integer convex and nonconvex optimization", JOTA, (2024)]

[M. De Santis, G. Eichfelder, J. Niebling, S. Rocktäschel; "Solving multiobjective mixed integer convex optimization problems", SIOPT, (2020)]

- ※ Look for efficient integer assignments:

branching on integer variables

[M. De Santis, G. Eichfelder, D. Patria, L. Warnow, "Using dual relaxations in multiobjective mixed-integer convex quadratic programming", JOGO, (2025)]

[M. De Santis, G. Eichfelder, "A decision space algorithm for multiobjective convex quadratic integer optimization", C & OR, (2021)]

...

`MObBQ`: a branch-and-bound method for multi-objective binary quadratic programs

Marianna De Santis, Lucas Létocart, Yue Zhang

UNIVERSITÀ DEGLI STUDI FIRENZE

UNIVERSITÉ SORBONNE PARIS NORD

M. De Santis, L. Létocart, Y. Zhang. *Quadratic Convex Reformulations for Multi-Objective Binary Quadratic Programming.* Optimization Online, 2025.

Multi-objective binary quadratic programs

problem formulation

$$
\begin{aligned}
\min_{x} \quad & (q_1(x), \dots, q_p(x))^\top \\
\text{s.t.} \quad & Ax \le b \\
& x_i \in \{0, 1\} \qquad\qquad i \in [n],
\end{aligned}
\tag{MO-BQP}
$$

where

$$q_j(x) = x^\top Q_j x + (c^j)^\top x, \; j \in [p]$$

Multi-objective binary quadratic programs

problem formulation

$$\begin{aligned} \min_{x} \quad & (q_1(x), \dots, q_p(x))^\top \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0, 1\} \qquad\qquad i \in [n], \end{aligned} \tag{MO-BQP}$$

where

$$q_j(x) = x^\top Q_j x + (c^j)^\top x, \; j \in [p]$$

with

- $Q_j \in \mathcal{S}^n, j \in [p]$ **not necessarily positive semidefinite**,

Multi-objective binary quadratic programs

problem formulation

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

where

$$q_j(x) = x^\top Q_j x + (c^j)^\top x, \; j \in [p]$$

with

- $Q_j \in \mathcal{S}^n, j \in [p]$ **not necessarily positive semidefinite**,
- $c^j \in \mathbb{R}^n, j \in [p]$,
- $A \in \mathbb{R}^{m \times n}, \;\; b \in \mathbb{R}^m$.

Examples of MO-BQPs

(MO-BQP) includes multi-objective combinatorial problems such as:

Examples of MO-BQPs

(MO-BQP) includes multi-objective combinatorial problems such as:

- **Multi-objective Max-Cut problem** (MO-MCP)

  Unconstrained MO-BQP

Examples of MO-BQPs

(MO-BQP) includes multi-objective combinatorial problems such as:

- **Multi-objective Max-Cut problem** (MO-MCP)

  Unconstrained MO-BQP

- **Multi-objective Quadratic Knapsack**

  in our numerical results, we consider the MultiObjective k-item Quadratic Knapsack problem (MO-kQKP)

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

Set $d = 0,\ r_d = \emptyset,\ \mathcal{L} := \{N^{r_d}\}$

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

Set $d = 0,\ r_d = \emptyset,\ \mathcal{L} := \{N^{r_d}\}$

While $\mathcal{L} \neq \emptyset$ **do**

Choose $N^{r_d} \in \mathcal{L}$, Update $\mathcal{L} \leftarrow \mathcal{L} \setminus \{N^{r_d}\}$

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

Set $d = 0,\ r_d = \emptyset,\ \mathcal{L} := \{N^{r_d}\}$

While $\mathcal{L} \neq \emptyset$ **do**

> Choose $N^{r_d} \in \mathcal{L}$, Update $\mathcal{L} \leftarrow \mathcal{L} \setminus \{N^{r_d}\}$
>
> Compute a lower bound set LB^{r_d} and evaluate `Prune(`N^{r_d}`)`

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

Set $d = 0,\ r_d = \emptyset,\ \mathcal{L} := \{N^{r_d}\}$

While $\mathcal{L} \neq \emptyset$ **do**

Choose $N^{r_d} \in \mathcal{L}$, Update $\mathcal{L} \leftarrow \mathcal{L} \setminus \{N^{r_d}\}$

Compute a lower bound set LB^{r_d} and evaluate `Prune(`N^{r_d}`)`

If (`not Prune(`N^{r_d}`)`) **then**

If (`CheckInt(`N^{r_d}`)`) **then**

Update S including $f(\bar{x})$ and keeping S a stable set

Update $\mathcal{U} = U(S)$

MO-bBQ scheme

Compute a starting stable set $S \subset \mathbb{R}^p$ and an upper bound set $\mathcal{U} = U(S)$

Set $d = 0,\ r_d = \emptyset,\ \mathcal{L} := \{N^{r_d}\}$

While $\mathcal{L} \neq \emptyset$ **do**

Choose $N^{r_d} \in \mathcal{L}$, Update $\mathcal{L} \leftarrow \mathcal{L} \setminus \{N^{r_d}\}$

Compute a lower bound set LB^{r_d} and evaluate `Prune(`N^{r_d}`)`

If (`not Prune(`N^{r_d}`)`) **then**

If (`CheckInt(`N^{r_d}`)`) **then**

Update S including $f(\bar{x})$ and keeping S a stable set

Update $\mathcal{U} = U(S)$

Else

Set $r^0_{d+1} = (r_d, 0),\ r^1_{d+1} = (r_d, 1)$

Update $\mathcal{L} \leftarrow \mathcal{L} \cup \{N^{r^0_{d+1}}, N^{r^1_{d+1}}\}$

End If

End While

Correctness of `MObBQ`

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

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

Correctness of `MObBQ`

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

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

Proposition

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

Correctness of `MObBQ`

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

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

Proposition

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

Proposition

`MObBQ` stops after a finite number of iterations returning the stable set S, that is the nondominated set of (MO-BQP).

Computation of lower bound sets at the nodes

Restricted continuous relaxation

$$
\begin{aligned}
\min \quad & (q_1(x), \ldots, q_p(x))^T \\
\text{s.t.} \quad & x_1 = r_1 \\
& x_2 = r_2 \\
& \quad \vdots \\
& x_d = r_d \\
& Ax \leq b \\
& x \in \mathbb{R}^n
\end{aligned}
$$

Computation of lower bound sets at the nodes

Restricted continuous relaxation

$$
\begin{aligned}
\min \quad & (q_1(x), \dots, q_p(x))^T \\
\text{s.t.} \quad & x_1 = r_1 \\
& x_2 = r_2 \\
& \vdots \\
& x_d = r_d \\
& Ax \le b \\
& x \in \mathbb{R}^n
\end{aligned}
\quad = \quad
\begin{aligned}
\min \quad & (q_1^r(x), \dots, q_p^r(x))^T \\
\text{s.t.} \quad & A^d x \le b^r \\
& x \in \mathbb{R}^{n-d}
\end{aligned}
$$

Quadratic Convex Reformulation[1]

single objective case

Consider the following single-objective binary quadratic problem

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

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

Consider the following single-objective binary quadratic problem

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

For any feasible point $x \in \{0,1\}^n$ it holds:

- $x_i^2 = x_i, \qquad i \in \{1,\dots,n\}$
- $(A_= x - b_=)^2 = (A_= x - b_=)^T (A_= x - b_=) = 0$

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

Introducing the parameters $\delta \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ problem BQP can be reformulated as

$$\begin{aligned} \min_x \quad & q_{\delta,\beta}(x) = q(x) + \sum_{i=1}^n \delta_i (x_i^2 - x_i) + \beta (A_= x - b_=)^2 \\ \text{s.t.} \quad & Ax \le b \qquad\qquad (\mathrm{QCR}_{(\delta,\beta)}) \\ & x_i \in \{0,1\} \quad i \in \{1,\dots,n\}. \end{aligned}$$

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

Introducing the parameters $\delta \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ problem BQP can be reformulated as

$$\begin{aligned} \min_{x} \quad & q_{\delta,\beta}(x) = q(x) + \textstyle\sum_{i=1}^{n} \delta_i (x_i^2 - x_i) + \beta (A_= x - b_=)^2 \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0,1\} \quad i \in \{1,\dots,n\}. \end{aligned} \qquad (\mathrm{QCR}_{(\delta,\beta)})$$

Note that $q_{\delta,\beta}(x)$ is a **quadratic function** depending on the matrix

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

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

Introducing the parameters $\delta \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ problem BQP can be reformulated as

$$\begin{aligned} \min_x \quad & q_{\delta,\beta}(x) = q(x) + \textstyle\sum_{i=1}^n \delta_i(x_i^2 - x_i) + \beta(A_= x - b_=)^2 \\ \text{s.t.} \quad & Ax \le b \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\mathrm{QCR}_{(\delta,\beta)}) \\ & x_i \in \{0,1\} \quad i \in \{1,\dots,n\}. \end{aligned}$$

$$\Downarrow$$

Any LB on $(\mathrm{QCR}_{(\delta,\beta)})$ **is a valid LB on** (BQP), for any $(\delta, \beta) \in \mathbb{R}^{n+1}$

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

look for the *best* $(\delta, \beta) \in \mathbb{R}^{n+1}$, **keeping $q_{\delta,\beta}$ convex:**

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

look for the *best* $(\delta, \beta) \in \mathbb{R}^{n+1}$, **keeping $q_{\delta,\beta}$ convex:**

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

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

single objective case

look for the *best* $(\delta, \beta) \in \mathbb{R}^{n+1}$, **keeping $q_{\delta,\beta}$ convex:**

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

(δ^*, β^*) are obtained as the **optimal dual variables of an SDP**

[1] Billionnet, Elloumi, and Plateau. “Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method.” Discret. Appl. Math. 157, 1185–1197 (2009).

Quadratic Convex Reformulation[1]

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

The optimal δ^* and β^* are the optimal dual variables associated with constraints (1) and (2).

[1] Billionnet, Elloumi, and Plateau. "Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method." Discret. Appl. Math. 157, 1185–1197 (2009).

Ignoring the constraints

The Unconstrained Quadratic Convex Reformulation (UQCR)

Ignoring $A_= x = b_=$, we obtain:

$$\begin{aligned} \min_x \quad & q_\delta(x) = q(x) + \sum_{i=1}^n \delta_i (x_i^2 - x_i) \\ \text{s.t.} \quad & Ax \le b \\ & x_i \in \{0, 1\} \qquad i \in \{1, \dots, n\}. \end{aligned} \tag{UQCR$_\delta$}$$

Ignoring $A_= x = b_=$, we obtain:

$$\begin{aligned} \min_x \quad & q_\delta(x) = q(x) + \sum_{i=1}^n \delta_i(x_i^2 - x_i) \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \{0,1\} \qquad\qquad i \in \{1,\ldots,n\}. \end{aligned} \qquad (\text{UQCR}_\delta)$$

The optimal $\delta^{u,*} \in \mathbb{R}^n$ is such that

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

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

$$\theta(\delta^{u,*}) \leq \theta(\delta^*, \beta^*)$$

Multi-objective Quadratic Convex Reformulation

We can compute $\delta_{0,j}^{u,*} \in \mathbb{R}^n$ and $(\delta_{0,j}^{*}, \beta_{0,j}^{*}) \in \mathbb{R}^{n+1}$ for each objective function and reformulate (MO-BQP) as:

Multi-objective Quadratic Convex Reformulation

We can compute $\delta_{0,j}^{u,*} \in \mathbb{R}^n$ and $(\delta_{0,j}^*, \beta_{0,j}^*) \in \mathbb{R}^{n+1}$ for each objective function and reformulate (MO-BQP) as:

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

$$
\begin{aligned}
\min_x \quad & ((q_1)_{(\delta_{0,1}^*, \beta_{0,1}^*)}(x), \dots, (q_p)_{(\delta_{0,p}^*, \beta_{0,p}^*)}(x)) \\
\text{s.t.} \quad & Ax \leq b \\
& x_i \in \{0,1\} \quad i \in [n]
\end{aligned}
\tag{MO-QCR}
$$

Multi-objective Quadratic Convex Reformulation

We can compute $\delta^{u,*}_{0,j} \in \mathbb{R}^n$ and $(\delta^*_{0,j}, \beta^*_{0,j}) \in \mathbb{R}^{n+1}$ for each objective function and reformulate (MO-BQP) as:

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

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

Proposition

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

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

Multi-objective Quadratic Convex Reformulation

At a generic node of the branch-and-bound tree, we have a vector of fixings $r_d \in \{0, 1\}^d$ and (MO-BQP)rd can be reformulated as:

Multi-objective Quadratic Convex Reformulation

At a generic node of the branch-and-bound tree, we have a vector of fixings $r_d \in \{0,1\}^d$ and (MO-BQP)rd can be reformulated as:

$$\begin{array}{ll} \min\limits_{x} & ((q_1^{r_d})_{\delta_1^{u,*}}(x), \ldots, (q_p^{r_d})_{\delta_p^{u,*}}(x)) \\ \text{s.t.} & A^d x \leq b^{r_d} \\ & x \in \{0,1\}^{n-d} \end{array} \qquad \text{(MO-UQCR}^{rd}\text{)}$$

$$\begin{array}{ll} \min\limits_{x} & ((q_1^{r_d})_{(\delta_1^{*},\beta_1^{*})}(x), \ldots, (q_p^{r_d})_{(\delta_p^{*},\beta_p^{*})}(x)) \\ \text{s.t.} & A^d x \leq b^{r_d} \\ & x \in \{0,1\}^{n-d} \end{array} \qquad \text{(MO-QCR}^{rd}\text{)}$$

Multi-objective Quadratic Convex Reformulation

At a generic node of the branch-and-bound tree, we have a vector of fixings $r_d \in \{0,1\}^d$ and (MO-BQP)rd can be reformulated as:

$$\begin{aligned} \min_x \quad & ((q_1^{r_d})_{\delta_1^{u,*}}(x), \ldots, (q_p^{r_d})_{\delta_p^{u,*}}(x)) \\ \text{s.t.} \quad & A^d x \le b^{r_d} \\ & x \in \{0,1\}^{n-d} \end{aligned} \qquad \text{(MO-UQCR}^{rd}\text{)}$$

$$\begin{aligned} \min_x \quad & ((q_1^{r_d})_{(\delta_1^*,\beta_1^*)}(x), \ldots, (q_p^{r_d})_{(\delta_p^*,\beta_p^*)}(x)) \\ \text{s.t.} \quad & A^d x \le b^{r_d} \\ & x \in \{0,1\}^{n-d} \end{aligned} \qquad \text{(MO-QCR}^{rd}\text{)}$$

Each multi-objective quadratic convex reformulation comes at the price of solving p semidefinite programs

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image sets at the node are **obtained by solving single-objective subproblems with $w \in W$**:

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image sets at the node are **obtained by solving single-objective subproblems with $w \in W$**:

$$\theta^{r_d}(w) := \min\{w^\top (q^{r_d})_{(\delta^*, \beta^*)}(x) \mid A^d x \leq b^{r_d}, x \in [0,1]^{n-d}\}$$

$$\theta^{r_d}_u(w) := \min\{w^\top (q^{r_d})_{\delta^{u,*}}(x) \mid A^d x \leq b^{r_d}, x \in [0,1]^{n-d}\}$$

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Let $W \subseteq \{w \in \mathbb{R}^p_+ \mid ||w||_1 = 1\}$ be a **finite set of nonnegative vectors** which includes all p unit vectors.

The supporting hyperplanes of the upper image sets at the node are **obtained by solving single-objective subproblems with $w \in W$**:

$$\theta^{r_d}(w) := \min\{w^\top (q^{r_d})_{(\delta^*,\beta^*)}(x) \mid A^d x \leq b^{r_d}, x \in [0,1]^{n-d}\}$$

$$\theta^{r_d}_u(w) := \min\{w^\top (q^{r_d})_{\delta^{u,*}}(x) \mid A^d x \leq b^{r_d}, x \in [0,1]^{n-d}\}$$

Note each subproblem is a **continuous convex quadratic program**

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Once the $|W|$ continuous single-objective convex quadratic subproblems are solved, the **lower bound sets** are then defined as:

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Once the $|W|$ continuous single-objective convex quadratic subproblems are solved, the **lower bound sets** are then defined as:

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

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

Computation of lower bound sets

Outer approximations of the upper image sets of the relaxations of MO-QCRrd and MO-UQCRrd

Once the $|W|$ continuous single-objective convex quadratic subproblems are solved, the **lower bound sets** are then defined as:

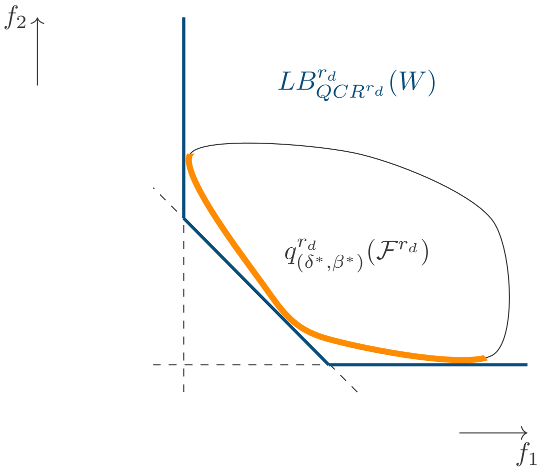
$$LB^{r_d}_{QCR^{r_d}}(W) := \bigcap_{w \in W} \{y \in \mathbb{R}^p \mid w^\top y \geq \theta^{r_d}(w)\}$$

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

The computation of a lower bound set at a node comes at the price of **solving p semidefinite programs and $|W|$ single-objective convex quadratic programs**

Lower bound sets as linear outer approximations

$W = \{(1,0),\ (0,1),\ (0.5,0.5)\}$



$\mathcal{F}^{r_d} = \{x \in [0,1]^{n-d} \mid A^d x \leq b^{r_d}\}$ is the restricted feasible set at N^{r_d}

Relaxations at the nodes

Comparing the lower bound sets

Proposition

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

Relaxations at the nodes

Comparing the lower bound sets

Proposition

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

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

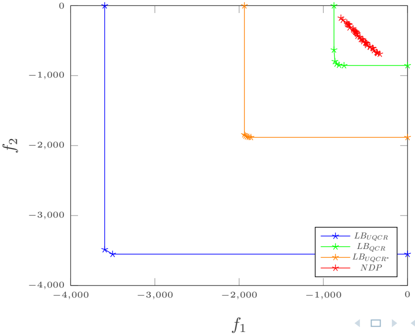
Relaxations at the nodes

Comparing the lower bound sets

Proposition

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

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



Advantage of static branching

Avoiding to solve p SDPs at each node

In our implementation of `MObBQ`, the order in which binary variables are fixed is **predetermined**

Advantage of static branching

Avoiding to solve p SDPs at each node

In our implementation of `MObBQ`, the order in which binary variables are fixed is **predetermined**

$$\Downarrow$$

at a generic level $d \in [n] \cup \{0\}$ of the search tree, the variables $x_1, \ldots, x_d$ are fixed.

Advantage of static branching

Avoiding to solve p SDPs at each node

In our implementation of `MObBQ`, the order in which binary variables are fixed is **predetermined**

$$\Downarrow$$

at a generic level $d \in [n] \cup \{0\}$ of the search tree, the variables $x_1, \ldots, x_d$ are fixed.

$$\Downarrow$$

The subproblems at the nodes belonging to the **same level** $d \in [n]$ share the **same matrices** Q_j^d, $j \in [p]$.

Computation of $\delta_j^{u,*}$, $j \in [p]$ in pre-processing

$p \times n$ SDPs to be solved in total

For each level $d = 0, \ldots, n-1$ and each $j \in [p]$ we address:

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

Computation of $\delta_j^{u,*}$, $j \in [p]$ in pre-processing

$p \times n$ SDPs to be solved in total

For each level $d = 0, \ldots, n-1$ and each $j \in [p]$ we address:

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

N.B. the linear term of the objective functions cannot be considered

Improved $\delta_j^{u,*}$

How to take into account $Ax \leq b$

If A and b have nonnegative entries

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

If A and b have nonnegative entries

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

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

Numerical results

We compared five different versions of $\texttt{MObBQ}$ algorithms:

- ◊ $\texttt{MObBQ}_{UQCR}$: (MO-UQCR)rd is adopted at every node
- ◊ $\texttt{MObBQ}_{UQCR^*}$: the *improved* UQCR is adopted at every node
- ◊ $\texttt{MObBQ}_{QCR+UQCR}$: (MO-QCR) is adopted at the root node and (MO-UQCR)rd is adopted at every other node
- ◊ $\texttt{MObBQ}_{QCR+UQCR^*}$: (MO-QCR) is adopted at the root node and the *improved* UQCR is adopted at every other node
- ◊ $\texttt{MObBQ}_{QCR^0}$ (or $\texttt{MObBQ}_{UQCR^0}$): (MO-QCR) (or (MO-UQCR)) is adopted to reformulate (MO-QCR)

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
  The implementation is available at
  https://github.com/Yue0925/MultiObjectiveAlgorithms.jl

Implementation details

◊ All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
The implementation is available at
`https://github.com/Yue0925/MultiObjectiveAlgorithms.jl`
...hopefully soon part of:

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
  The implementation is available at
  https://github.com/Yue0925/MultiObjectiveAlgorithms.jl
- We invoke CSDP.JL solver v.6.2.0 and GUROBI.JL solver v12.0.0 (academic license).

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
  The implementation is available at
  https://github.com/Yue0925/MultiObjectiveAlgorithms.jl
- We invoke CSDP.JL solver v.6.2.0 and GUROBI.JL solver v12.0.0 (academic license).
- The number of hyperplanes considered for the computation of LBS is set to $|W| = p + 1$.

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
  The implementation is available at
  https://github.com/Yue0925/MultiObjectiveAlgorithms.jl
- We invoke CSDP.JL solver v.6.2.0 and GUROBI.JL solver v12.0.0 (academic license).
- The number of hyperplanes considered for the computation of LBS is set to $|W| = p + 1$.
- The starting stable set S (and then the starting upperbound set) is set as the empty set.

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2.
  The implementation is available at
  https://github.com/Yue0925/MultiObjectiveAlgorithms.jl
- We invoke CSDP.JL solver v.6.2.0 and GUROBI.JL solver v12.0.0 (academic license).
- The number of hyperplanes considered for the computation of LBS is set to $|W| = p + 1$.
- The starting stable set S (and then the starting upperbound set) is set as the empty set.
- The time limit (CPU) is set to 1800 seconds.

Implementation details

- All versions of `MO-bBQ` are implemented in JULIA v.1.11.2. The implementation is available at https://github.com/Yue0925/MultiObjectiveAlgorithms.jl
- We invoke CSDP.JL solver v.6.2.0 and GUROBI.JL solver v12.0.0 (academic license).
- The number of hyperplanes considered for the computation of LBS is set to $|W| = p + 1$.
- The starting stable set S (and then the starting upperbound set) is set as the empty set.
- The time limit (CPU) is set to 1800 seconds.
- We compare with the ε-constraint method on BO instances.

Results on bi-objective max-cut instances

comparison with the ϵ-constraint method available in `JuMP` at
`https://jump.dev/JuMP.jl/stable/packages/MultiObjectiveAlgorithms`

n	density%	$\lfloor$#NDP$\rfloor$	ϵ-constraint		$\texttt{MObBQ}_{UQCR^0}$			$\texttt{MObBQ}_{UQCR}$		
			solved	time(s)	solved	time(s)	#nodes	solved	time(s)	#nodes
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	-	-

Results on tri-objective max cut instances

applying UQCR at every node works better than a one-time reformulation

n	density%	$\lfloor$#NDP$\rceil$	$\texttt{MObBQ}_{UQCR^0}$			$\texttt{MObBQ}_{UQCR}$		
			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	-	-

Results on multi-objective k-item quadratic knapsack

problem formulation

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

where $f_j(x) = \sum_{i=1}^{n} \sum_{\ell=1}^{n} (Q_j)_{i\ell} x_i x_l$.

Results on bi-objective kQKP

We employ the single-objective instance generator described in [Ceselli et al., MPC (2022)]

n	density%	$\lfloor$#NDP$\rfloor$	$\texttt{MObBQ}_{QCR^0}$			$\texttt{MObBQ}_{UQCR}$			$\texttt{MObBQ}_{QCR+UQCR}$			$\texttt{MObBQ}_{UQCR^*}$			$\texttt{MObBQ}_{QCR+UQCR^*}$		
			sol	time(s)	#nodes	sol	time(s)	#nodes	sol	time(s)	#nodes	sol	time(s)	#nodes	sol	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

Results on bi-objective kQKP

Comparison with the ϵ-constraint method available in `JuMP` at `https://jump.dev/JuMP.jl/stable/packages/MultiObjectiveAlgorithms`

n	density%	$\lfloor$#NDP$\rfloor$	ϵ-constraint		MObBQ$_{UQCR}$	
			sol	time(s)	sol	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

Take-home messages

Take-home messages

- MOMINLPs are challenging (and engaging!) problems.

Take-home messages

- MOMINLPs are challenging (and engaging!) problems.
- Single-objective branch-and-bound concepts have been extended through the *enclosure*, the *lower bound sets* and *upper bound sets* concepts.

Take-home messages

- MOMINLPs are challenging (and engaging!) problems.
- Single-objective branch-and-bound concepts have been extended through the *enclosure*, the *lower bound sets* and *upper bound sets* concepts.
- The nondominated set can be *exactly* retrieved for:
  - MO integer nonlinear problems (it is a finite set);
  - BO mixed-integer linear problems (the aim is detecting the *extreme nondominated points*).

Take-home messages

- MOMINLPs are challenging (and engaging!) problems.
- Single-objective branch-and-bound concepts have been extended through the *enclosure*, the *lower bound sets* and *upper bound sets* concepts.
- The nondominated set can be *exactly* retrieved for:
  - MO integer nonlinear problems (it is a finite set);
  - BO mixed-integer linear problems (the aim is detecting the *extreme nondominated points*).
- The quadratic convex reformulation paradigm can be extended to the multi-objective setting to define promising branch-and-bound methods.

Take-home messages

- MOMINLPs are challenging (and engaging!) problems.
- Single-objective branch-and-bound concepts have been extended through the *enclosure*, the *lower bound sets* and *upper bound sets* concepts.
- The nondominated set can be *exactly* retrieved for:
  - MO integer nonlinear problems (it is a finite set);
  - BO mixed-integer linear problems (the aim is detecting the *extreme nondominated points*).
- The quadratic convex reformulation paradigm can be extended to the multi-objective setting to define promising branch-and-bound methods.

Thanks for your attention!