Solution of integer quadratic programs through quadratic convex reformulation

Amélie Lambert

CEDRIC-ENSIIE

February 16, 2010
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
Presentation of the problem

\[
\begin{align*}
\text{(QP)} & \quad \text{Min} \quad f(x) = x^T Qx + c^T x \\
& \quad \text{s.t} \quad \sum_i a_{ri} x_i = b_r \quad \text{m equalities} \\
& \quad \sum_i d_{si} x_i \leq e_s \quad \text{p inequalities} \\
& \quad 0 \leq x_i \leq u_i \\
& \quad x_i \text{ integer} \quad \text{n variables}
\end{align*}
\]
Presentation of the problem

\[\begin{align*}
\text{Min} & \quad f(x) = x^T Q x + c^T x \\
\text{s.t} & \quad \sum_{i} a_{ri} x_i = b_r \quad \text{m equalities} \\
& \quad \sum_{i} d_{si} x_i \leq e_s \quad \text{p inequalities} \\
& \quad 0 \leq x_i \leq u_i \\
& \quad x_i \text{ integer} \quad \text{n variables}
\end{align*}\]

- **Double difficulty:**
  1. Integrity of the variables
  2. Non convexity of \( f(x) \)
Presentation of the problem

\[
\begin{align*}
\text{Min} & \quad f(x) = x^T Q x + c^T x \\
\text{s.t} & \quad \sum_{i} a_{ri} x_i = b_r \quad \text{m equalities} \\
& \quad \sum_{i} d_{si} x_i \leq e_s \quad \text{p inequalities} \\
& \quad 0 \leq x_i \leq u_i \\
& \quad x_i \text{ integer} \quad \text{n variables}
\end{align*}
\]

- Double difficulty:
  1. Integrity of the variables
  2. Non convexity of \( f(x) \)

- Existence of algorithms and solvers when \( f(x) \) is a convex function.
Presentation of the problem

Approach: Reformulate \((QP)\) into an equivalent program
Presentation of the problem

Approach:
1. Reformulate \((QP)\) into an **equivalent** program
2. Which objective function is **quadratic convex**.
   \(\Rightarrow\) the continuous relaxation become polynomial.
Presentation of the problem

Approach:
1. Reformulate \((QP)\) into an equivalent program

2. which objective function is quadratic convex.
   \(\Rightarrow\) the continuous relaxation become polynomial.

3. Submit the reformulated program to a solver.
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
Previous works : case 0-1

1. unconstrained quadratic programming :

   - The smallest eigenvalue method (Hammer, Rubin - 1970).
Previous works: case 0-1

1. unconstrained quadratic programming:

   - The smallest eigenvalue method (Hammer, Rubin - 1970).
   - The method UQCR (Unconstrained Quadratic Convex Reformulation) (Billionnet, Elloumi - 2007).
Previous works : case 0-1

1 unconstrained quadratic programming :

   ▶ The smallest eigenvalue method (Hammer, Rubin - 1970).

   ▶ The method UQCR (Unconstrained Quadratic Convex Reformulation) (Billionnet, Elloumi - 2007).

2 equality constrained quadratic programming :

   ▶ The method QCR (Quadratic Convex Reformulation) (Billionnet, Elloumi, Plateau - 2008).
A simple quadratic convex reformulation: the method NC (Naive Convexification)
The method NC

Idea: Perturb each diagonal term of $Q$ with the smallest eigenvalue of $Q$:

$\lambda_{\text{min}}(Q)$. 
The method NC

Idea : Perturb each diagonal term of $Q$ with the smallest eigenvalue of $Q : \lambda_{\text{min}}(Q)$.

Let $f_{\lambda_{\text{min}}(Q)}(x, v) = f(x) - \lambda_{\text{min}}(Q) \sum_{i=1}^{n} (x_i^2 - v_i)$
The method NC

Idea: Perturb each diagonal term of $Q$ with the smallest eigenvalue of $Q$: $\lambda_{\text{min}}(Q)$.

Let $f_{\lambda_{\text{min}}(Q)}(x, v) = f(x) - \lambda_{\text{min}}(Q) \sum_{i=1}^{n} (x_i^2 - v_i)$

We want $f_{\lambda_{\text{min}}(Q)}(x, v) = f(x)$ on the domain of $(x, v)$
The method NC

Idea: Perturb each diagonal term of $Q$ with the smallest eigenvalue of $Q$: $\lambda_{\text{min}}(Q)$.

Let $f_{\lambda_{\text{min}}(Q)}(x, v) = f(x) - \lambda_{\text{min}}(Q) \sum_{i=1}^{n} (x_i^2 - v_i)$

We want $f_{\lambda_{\text{min}}(Q)}(x, v) = f(x)$ on the domain of $(x, v)$

Linearization: $v_i = x_i^2$:

$$
\begin{align*}
  x_i &= \sum_{k=0}^{u_i} kr_{ik} \\
  \sum_{k=0}^{u_i} r_{ik} &= 1 \\
  v_i &= \sum_{k=0}^{u_i} k^2 r_{ik} \\
  r_{ik} &\in \{0, 1\}
\end{align*}
$$
Example

4 variables, 1 equality constraint and 1 inequality constraint:

\[
\begin{align*}
\text{Min} \quad & f(x) = x^T \begin{pmatrix}
5 & -7 & -6 & -1 \\
-7 & 3 & -8 & -18 \\
-6 & -8 & -17 & 10 \\
-1 & -18 & 10 & 3
\end{pmatrix} x + \begin{pmatrix}
-5 \\
-11 \\
4 \\
1
\end{pmatrix}^T x \\
(\text{QP}_e) \quad & s.t. \quad 3x_1 + 19x_2 + 18x_3 + 11x_4 = 255 \\
& \quad 11x_1 + 13x_2 + 8x_3 + x_4 \leq 165 \\
& \quad 0 \leq x_i \leq 10 \quad i \in \{1, \ldots, 4\} \\
& \quad x_i \text{ integer} \quad i \in \{1, \ldots, 4\}
\end{align*}
\]

optimal value: -2552.
Example: NC

With the method NC, we perturb the $Q$ matrix as follows:

$$
\begin{pmatrix}
5 - \lambda_{\text{min}}(Q) & -7 & -6 & -1 \\
-7 & 3 - \lambda_{\text{min}}(Q) & -8 & -18 \\
-6 & -8 & -17 - \lambda_{\text{min}}(Q) & 10 \\
-1 & -18 & 10 & 3 - \lambda_{\text{min}}(Q)
\end{pmatrix}
$$

where $\lambda_{\text{min}}(Q) = -22.64$. 
Example: NC

<table>
<thead>
<tr>
<th>optimum</th>
<th>-2552</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb var</td>
<td>NC</td>
</tr>
<tr>
<td>nb const</td>
<td>52</td>
</tr>
<tr>
<td>time (s)</td>
<td>14</td>
</tr>
<tr>
<td>cont relax</td>
<td>0.01</td>
</tr>
<tr>
<td>gap</td>
<td>-3991.85</td>
</tr>
<tr>
<td></td>
<td>56.42 %</td>
</tr>
</tbody>
</table>
The quadratic convex reformulation IQCR
(Integer Quadratic Convex Reformulation)
Step 1: Define a family of equivalent problems to \((QP)\).
The IQCR method (Billionnet, Elloumi, Lambert-2009)

Step 1 : Define a family of equivalent problems to \((QP)\).

Step 2 : Choose within this family the problem \((QP')\) such that :

1. The objective function of \((QP')\) is \textbf{convex}.

2. The bound obtained by continuous relaxation of \((QP')\) is as \textbf{close} as possible.
A new family of convex reformulation of $(QP)$

Idea: Perturb each term of $Q$ by a specific coefficient using a matrix parameter $\beta$, and use equality constraints to perturb the objective function using a scalar parameter $\alpha$. 
A new family of convex reformulation of \((QP)\)

Idea: Perturb each term of \(Q\) by a specific coefficient using a matrix parameter \(\beta\), and use equality constraints to perturb the objective function using a scalar parameter \(\alpha\).

Let \(f_{\alpha,\beta}(x, y) = f(x) + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(x_i x_j - y_{ij}) + \alpha \sum_{r=1}^{m} \left( \sum_{i=1}^{n} a_{ri} x_i - b_r \right)^2\)
A new family of convex reformulation of \((QP)\)

Idea: Perturb each term of \(Q\) by a specific coefficient using a matrix parameter \(\beta\), and use equality constraints to perturb the objective function using a scalar parameter \(\alpha\).

\[
\text{Let } f_{\alpha,\beta}(x, y) = f(x) + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} (x_i x_j - y_{ij}) + \alpha \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2
\]

We want \(f_{\alpha,\beta}(x, y) = f(x)\) on the domain of \((x, y)\):
A new family of convex reformulation of \((QP)\)

**Idea:** Perturb each term of \(Q\) by a specific coefficient using a matrix parameter \(\beta\), and use equality constraints to perturb the objective function using a scalar parameter \(\alpha\).

Let \(f_{\alpha, \beta}(x, y) = f(x) + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(x_i x_j - y_{ij}) + \alpha \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri}x_i - b_r)^2\)

We want \(f_{\alpha, \beta}(x, y) = f(x)\) on the domain of \((x, y)\):

\[\alpha \sum_{r=1}^{m} \left( \sum_{i=1}^{n} a_{ri}x_i - b_r \right)^2 = 0.\]
A new family of convex reformulation of (QP)

Idea: Perturb each term of $Q$ by a specific coefficient using a matrix parameter $\beta$, and use equality constraints to perturb the objective function using a scalar parameter $\alpha$.

Let $f_{\alpha,\beta}(x, y) = f(x) + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(x_i x_j - y_{ij}) + \alpha \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2$

We want $f_{\alpha,\beta}(x, y) = f(x)$ on the domain of $(x, y)$:

1. $\alpha \sum_{r=1}^{m} (\sum_{i=1}^{n} a_{ri} x_i - b_r)^2 = 0$.

2. $\sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(x_i x_j - y_{ij}) = 0$ if $y_{ij} = x_i x_j$. 
Linearization of $y_{ij} = x_i x_j$ (BIL(Binary Integer Linearization) (Billionnet, Elloumi, Lambert-2008)

Binary decomposition of each variable $x_j$

\[
x_j = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk} \text{ avec } t_{jk} \in \{0, 1\}
\]
Linearization of $y_{ij} = x_i x_j$ (BIL (Binary Integer Linearization) (Billionnet, Elloumi, Lambert-2008)

1. Binary decomposition of each variable $x_j$

$$x_j = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk} \text{ avec } t_{jk} \in \{0, 1\}$$

2. $x_i x_j = x_i (\sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk}) = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k x_i t_{jk},$
Linearization of $y_{ij} = x_i x_j$ (BIL(Binary Integer Linearization) (Billionnet, Elloumi, Lambert-2008)

1. Binary decomposition of each variable $x_j$

$$x_j = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk} \text{ avec } t_{jk} \in \{0, 1\}$$

2. $x_i x_j = x_i \left( \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk} \right) = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k x_i t_{jk}$

3. Linearization: $z_{ijk} = x_i t_{jk}$

$$\begin{cases} z_{ijk} \leq t_{jk} u_i \\ z_{ijk} \leq x_i \\ z_{ijk} \geq x_i - u_i(1 - t_{jk}) \\ z_{ijk} \geq 0 \end{cases} \iff x_i t_{jk} = z_{ijk}$$
Adding valid inequalities

We add the following valid inequalities:
Adding valid inequalities

We add the following valid inequalities:

\[ y_{ij} = y_{ji}, \text{ because } y_{ij} = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k x_i t_{jk} \text{ and } y_{ji} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k x_j t_{ik} \]
Adding valid inequalities

We add the following valid inequalities:

1. $y_{ij} = y_{ji}$, because $y_{ij} = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k x_i t_{jk}$ and $y_{ji} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k x_j t_{ik}$

2. $y_{ij} \geq u_i x_j + u_j x_i - u_i u_j$, follow from $(x_i - u_i)(x_j - u_j) \geq 0$. 
Adding valid inequalities

We add the following valid inequalities:

- $y_{ij} = y_{ji}$, because $y_{ij} = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k x_i t_{jk}$ and $y_{ji} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k x_j t_{ik}$

- $y_{ij} \geq u_i x_j + u_j x_i - u_i u_j$, follow from $(x_i - u_i)(x_j - u_j) \geq 0$.

- $y_{ii} \geq x_i$, because $x_i^2 \geq x_i$. 
IQCR - Step 1: A family of equivalent problems to (QP)

\[ (QP) \begin{cases} 
\text{Min} & f(x) \\
\text{s.t.} & Ax = b \\
& Dx \leq e \\
& 0 \leq x_i \leq u_i \\
& x_i \text{ integer} 
\end{cases} \iff \begin{cases} 
\text{Min} & f_{\alpha,\beta}(x, y) \\
\text{s.t.} & Ax = b \\
& Dx \leq e \\
& 0 \leq x_i \leq u_i \\
& x_i \text{ integer} \\
& M \leq x_j \leq M \\
& z_{ijk} \leq t_{jk} u_i \\
& z_{ijk} \leq x_i \\
& z_{ijk} \geq x_i - u_i (1 - t_{jk}) \\
& z_{ijk} \geq 0 \\
& y_{ij} = \sum_{k=0}^{[\log(u_j)]} 2^k t_{jk} \\
& y_{ij} \geq u_i x_j + u_j x_i - u_i u_j \\
& y_{ii} \geq x_i \\
& y_{ij} = y_{ji} \\
& t_{jk} \in \{0, 1\} 
\end{cases} \]
**IQCR - Step 1 : A family of equivalent problems to \((QP)\)**

\[(QP)\begin{array}{l}
\text{Min} \quad f(x) \\
\text{s.t.} \quad Ax = b \\
\phantom{\text{s.t.}} \quad Dx \leq e \\
\phantom{\text{s.t.}} \quad 0 \leq x_i \leq u_i \\
\phantom{\text{s.t.}} \quad x_i \text{ integer}
\end{array}\quad \iff \quad \begin{array}{l}
\text{Min} \quad f_{\alpha, \beta}(x, y) \\
\text{s.t.} \quad Ax = b \\
\phantom{\text{s.t.}} \quad Dx \leq e \\
\phantom{\text{s.t.}} \quad 0 \leq x_i \leq u_i \\
\phantom{\text{s.t.}} \quad [\log(u_j)] \\
\phantom{\text{s.t.}} \quad x_j = \sum_{k=0}^{\log(u_j)} 2^k t_{jk} \\
\phantom{\text{s.t.}} \quad z_{ijk} \leq t_{jk} u_i \\
\phantom{\text{s.t.}} \quad z_{ijk} \leq x_i \\
\phantom{\text{s.t.}} \quad z_{ijk} \geq x_i - u_i(1 - t_{jk}) \\
\phantom{\text{s.t.}} \quad z_{ijk} \geq 0 \\
\phantom{\text{s.t.}} \quad [\log(u_j)] \\
\phantom{\text{s.t.}} \quad y_{ij} = \sum_{k=0}^{\log(u_j)} 2^k z_{ijk} \\
\phantom{\text{s.t.}} \quad y_{ij} \geq u_i x_j + u_j x_i - u_i u_j \\
\phantom{\text{s.t.}} \quad y_{ii} \geq x_i \\
\phantom{\text{s.t.}} \quad y_{ij} = y_{ji} \\
\phantom{\text{s.t.}} \quad t_{jk} \in \{0, 1\}
\end{array}\]

**Remark** : there exists \(\alpha\) and \(\beta\) such that \(f_{\alpha, \beta}(x, y)\) is convex.
IQCR – Step 2:

Choose $\alpha$ and $\beta$ such that:

1. $f_{\alpha,\beta}(x, y)$ is convex.

2. The value of $(QP_{\alpha,\beta})$ is maximal.
Theorem (a projection property)

\[
\begin{align*}
\text{Min} & \quad f_{\alpha, \beta}(x, y) \\
\text{s.t.} & \quad Ax = b \\
& \quad Dx \leq e \\
& \quad 0 \leq x_i \leq u_i \\
& \quad \lfloor \log(u_i) \rfloor \\
& \quad x_j = \sum_{k=0}^{\lfloor \log(u_j) \rfloor} 2^k t_{jk} \\
& \quad z_{ijk} \leq t_{jk} u_i \\
& \quad z_{ijk} \leq x_i \\
& \quad z_{ijk} \geq x_i - u_i(1 - t_{jk}) \\
& \quad z_{ijk} \geq 0 \\
& \quad y_{ij} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k z_{ijk} \\
& \quad y_{ij} \geq u_i x_j + u_j x_i - u_i u_j \\
& \quad y_{ii} \geq x_i \\
& \quad y_{ij} = y_{ji} \\
& \quad 0 \leq t_{jk} \leq 1
\end{align*}
\]

Projection of the polyhedra of \((QP_{\alpha, \beta})\) on \(x\) and \(y = \) polyhedra of \((P_{\alpha, \beta})\)
Theorem (a projection property)

\[ \begin{align*}
\text{Min} & \quad f_{\alpha, \beta}(x, y) \\
\text{s.t.} & \quad Ax = b \\
& \quad Dx \leq e \\
& \quad 0 \leq x_i \leq u_i \\
& \quad \lfloor \log(u_j) \rfloor \\
& \quad x_j = \sum_{k=0} \lfloor \log(u_j) \rfloor 2^k t_{jk} \\
& \quad z_{ijk} \leq t_{jk} u_i \\
& \quad z_{ijk} \leq x_i \\
& \quad z_{ijk} \geq x_i - u_i(1 - t_{jk}) \\
& \quad z_{ijk} \geq 0 \\
& \quad \lfloor \log(u_j) \rfloor \\
& \quad y_{ij} = \sum_{k=0} \lfloor \log(u_j) \rfloor 2^k z_{ijk} \\
& \quad y_{ij} \geq u_i x_j + u_j x_i - u_i u_j \\
& \quad y_{ii} \geq x_i \\
& \quad y_{ij} = y_{ji} \\
& \quad 0 \leq t_{jk} \leq 1
\end{align*} \]

Projection of the polyhedra of \( (QP_{\alpha, \beta}) \) on \( x \) and \( y = \) polyhedra of \( (P_{\alpha, \beta}) \)

Theorical and practical interest for IQCR.
Theorem (computation of optimal $\alpha$ and $\beta$) : IQCR

The optimal values of $\alpha$ and $\beta$ that:

1. make $f_{\alpha,\beta}(x, y)$ convex
2. maximize the value of $(QP_{\alpha,\beta})$

can be computed by a semi-definite relaxation of $(QP)$.

\[
\begin{align*}
\text{Min} \quad & f(X, x) = \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} X_{ij} + \sum_{i=1}^{n} c_{i} x_{i} \\
\text{s.t.} \quad & \sum_{i} a_{ri} x_{i} = b_{r} \\
& \sum_{i} d_{si} x_{i} \leq e_{s} \\
& \sum_{r} \left( \sum_{i} \sum_{j} a_{ri} a_{rj} X_{ij} - 2 \sum_{i} a_{ri} b_{r} x_{i} \right) = -\sum_{r} (b_{r}^{2}) \quad \leftarrow \alpha \\
& X_{ij} \geq x_{i} \quad \forall i \\
& X_{ij} \leq u_{j} x_{i} \quad \forall i, j \\
& X_{ij} \leq u_{i} x_{j} \quad \forall i, j \\
& X_{ij} \geq u_{j} x_{i} + u_{i} x_{j} - u_{i} u_{j} \quad \forall i, j \\
& X_{ij} \geq 0 \quad \forall i, j \\
& \left( \begin{array}{cc}
1 & x \\
x^{T} & X \\
\end{array} \right) \succeq 0 \\
x \in \mathbb{R}^{n} \quad X \in \mathbb{R}^{n \times n}
\end{align*}
\]
Recap on IQCR:

Property (Quality of the SDP bound at the root of the B&B)

Branch & Bound based on the continuous quadratic convex relaxation:

\[ v(QP_{\alpha,\beta}) = v(SDP) \]
Recap on IQCR:

Property (Quality of the SDP bound at the root of the B&B)

Branch & Bound based on the continuous quadratic convex relaxation:

\[ v(QP_{\alpha,\beta}) = v(SDP) \]

Property (feasible solutions of the dual of \((SDP)\))

\( \forall (\alpha, \beta) \) dual solution of \((SDP)\):

1. \((QP_{\alpha,\beta})\) is equivalent to \((QP)\).

2. \( f_{\alpha,\beta}(x, y) \) is convex.
Example (recall)

4 variables, 1 equality constraint and 1 inequality constraint :

\[
\begin{align*}
\text{Min } f(x) &= x^T \begin{pmatrix}
5 & -7 & -6 & -1 \\
-7 & 3 & -8 & -18 \\
-6 & -8 & -17 & 10 \\
-1 & -18 & 10 & 3
\end{pmatrix} x + \begin{pmatrix}
-5 \\
-11 \\
4 \\
1
\end{pmatrix}^T x \\
\text{s.t. } 3x_1 + 19x_2 + 18x_3 + 11x_4 &= 255 \\
11x_1 + 13x_2 + 8x_3 + x_4 &\leq 165 \\
0 \leq x_i &\leq 10, \quad i \in \{1, \ldots, 4\} \\
x_i \text{ integer, } \quad i \in \{1, \ldots, 4\}
\end{align*}
\]

optimal value : -2552.
Example : IQCR

With the method IQCR, we perturb the $Q$ matrix as follows :

$$
\begin{pmatrix}
5 + \beta_{11} + 9\alpha & -7 + \beta_{12} + 57\alpha & -6 + \beta_{13} + 54\alpha & -1 + \beta_{14} + 33\alpha \\
-7 + \beta_{12} + 57\alpha & 3 + \beta_{22} + 361\alpha & -8 + \beta_{23} + 342\alpha & -18 + \beta_{24} + 209\alpha \\
-6 + \beta_{13} + 54\alpha & -8 + \beta_{23} + 342\alpha & -17 + \beta_{33} + 324\alpha & 10 + \beta_{34} + 198\alpha \\
-1 + \beta_{14} + 33\alpha & -18 + \beta_{24} + 209\alpha & 10 + \beta_{34} + 198\alpha & 3 + \beta_{44} + 121\alpha
\end{pmatrix}
$$

where $\alpha = 2090.76$, and $\beta_{11} = \beta_{12} = \beta_{13} = \beta_{14} = \beta_{22} = \beta_{23} = \beta_{24} = 0$, $\beta_{33} = 24.45$, $\beta_{34} = -6.80$, $\beta_{44} = 4.92$. 
Example : IQCR

<table>
<thead>
<tr>
<th>optimum</th>
<th>-2552</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQCR</td>
<td></td>
</tr>
<tr>
<td>nb var</td>
<td>100</td>
</tr>
<tr>
<td>nb cont</td>
<td>246</td>
</tr>
<tr>
<td>time (s)</td>
<td>0.004</td>
</tr>
<tr>
<td>cont relax</td>
<td>-2808.77</td>
</tr>
<tr>
<td>gap</td>
<td>10.06 %</td>
</tr>
</tbody>
</table>
An extension of IQCR

Extension of IQCR to the mixed-integer case: MIQCR (Mixed Integer Quadratic Convex Reformulation)

- **Difficulty**: We cannot linearize the product of two continuous variables.
An extension of IQCR

Extension of IQCR to the mixed-integer case: MIQCR (Mixed Integer Quadratic Convex Reformulation)

- **Difficulty**: We can not linearize the product of two continuous variables.

- **Assumption**: The sub-function of the products of continuous variables is convex.
An extension of IQCR

Extension of IQCR to the mixed-integer case: MIQCR (Mixed Integer Quadratic Convex Reformulation)

- **Difficulty**: We cannot linearize the product of two continuous variables.

- **Assumption**: The sub-function of the products of continuous variables is convex.

- **Method**: The perturbation $\beta_{ij}(x_i x_j - y_{ij})$ is not considered if $x_i$ and $x_j$ are reals.
An improvement of IQCR

**Improvement : Perturb $f(x)$ with inequality constraints**

1. Transform the $p$ inequality constraints into equality constraints using $p$ continuous slack variables.

2. Apply MIQCR

⇒ Perturbation of the objective function with inequality constraints.
Example

4 integer variables, 1 continuous slack variable, and 2 equality constraints.

\[
\begin{align*}
\text{Min } f(x) &= x^T \begin{pmatrix} 5 & -7 & -6 & -1 & 0 \\ -7 & 3 & -8 & -18 & 0 \\ -6 & -8 & -17 & 10 & 0 \\ -1 & -18 & 10 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} -5 \\ -11 \\ 4 \\ 1 \\ 0 \end{pmatrix}^T x \\
\text{s.t. } &3x_1 + 19x_2 + 18x_3 + 11x_4 = 255 \\
&11x_1 + 13x_2 + 8x_3 + x_4 + s = 165 \\
&0 \leq x_i \leq 10 \\
&0 \leq s \leq 165 \\
&x_i \text{ integer} \\
&s \text{ réel}
\end{align*}
\]

\((QP_e')\)
An interesting restriction of IQCR: the method CQCR
(Compact Quadratic Convex Convex Reformulation)

Difference: \[ \beta_{ij}(x_i x_j - y_{ij}) \]
An interesting restriction of IQCR : the method CQCR (Compact Quadratic Convex Convex Reformulation)

Difference : 

\[ \text{In IQCR : } \beta_{ij}(x_i x_j - y_{ij}) \]
\[ \text{In CQCR : } \beta_i(x_i^2 - w_i) \]
An interesting restriction of IQCR: the method CQCR
(Compact Quadratic Convex Convex Reformulation)

Difference:

In IQCR: $\beta_{ij}(x_i x_j - y_{ij})$

In CQCR: $\beta_i(x_i^2 - w_i)$

- Less constraints, less variables, but a bound not as close.
An interesting restriction of IQCR: the method CQCR (Compact Quadratic Convex Convex Reformulation)

Difference:

\[ \text{In IQCR: } \beta_{ij}(x_i x_j - y_{ij}) \]
\[ \text{In CQCR: } \beta_i(x_i^2 - w_i) \]

- Less constraints, less variables, but a bound not as close.
- Practical interest, because CQCR can accelerate the computation time.
Example : NC, IQCR, MIQCR, et CQCR

<table>
<thead>
<tr>
<th>optimum</th>
<th>-2552</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>IQCR</td>
</tr>
<tr>
<td>nb var</td>
<td>52</td>
</tr>
<tr>
<td>nb cont</td>
<td>14</td>
</tr>
<tr>
<td>time (s)</td>
<td>0.01</td>
</tr>
<tr>
<td>cont relax</td>
<td>-3991.82</td>
</tr>
<tr>
<td>gap</td>
<td>56.42%</td>
</tr>
</tbody>
</table>
Interpretation of these convexifications for 0-1 quadratic programming ($u_i = 1$)
Interpretation of these convexifications for 0-1 quadratic programming \((u_i = 1)\)

- NC correspond to the smallest eigenvalue method.
Interpretation of these convexifications for 0-1 quadratic programming ($u_i = 1$)

- NC correspond to the smallest eigenvalue method.
- CQCR correspond to the method QCR
Interpretation of these convexifications for 0-1 quadratic programming ($u_i = 1$)

- NC correspond to the smallest eigenvalue method.
- CQCR correspond to the method QCR
- IQCR constitute an improvement of the method QCR
Interpretation of these convexifications for 0-1 quadratic programming \( (u_i = 1) \)

- NC correspond to the smallest eigenvalue method.
- CQCR correspond to the method QCR.
- IQCR constitute an improvement of the method QCR.
- MIQCR also constitute an improvement of the method QCR.
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
Property:

Let \( (P^e_{\alpha,\beta}) \) be the following problem:

\[
\begin{align*}
\text{Min} & \quad f_{\alpha,\beta}(x, y) \\
\text{s.t} & \quad x \in \mathbb{R}^n \\
& \quad Ax = b \\
& \quad Dx \leq e \\
& \quad y_{ij} \geq 0 \\
& \quad y_{ij} \geq u_i x_j + u_j x_i - u_i u_j \\
& \quad y_{ij} \leq u_i x_j \\
& \quad y_{ij} \leq u_j x_i \\
& \quad y_{ii} \geq x_i \\
& \quad y_{ij} = y_{ji} \\
& \quad x_i \in \mathbb{N}
\end{align*}
\]

An optimal solution of \((QP)\), i.e. of \((QP_{\alpha,\beta})\), can be obtained by solving the problem \(P^e_{\alpha,\beta}\) with the additional constraints \(x_i x_j = y_{ij}\).
A Branch & Bound algorithm (1)

Approach: At each node of the tree solve $(P^e_{\alpha,\beta})$, then branch in the following way:

Let $(\bar{x}, \bar{y})$ be the solution at the current node:

1. If $\forall (i, j) \in I^2$, $\bar{x}_i \bar{x}_j = \bar{y}_{ij}$, then $\bar{x}$ is a feasible solution to $(QP)$ and $f(\bar{x}) = f_{\alpha,\beta}(\bar{x}, \bar{y})$.
2. If $\exists (i, j) \in I^2$, such that $\bar{x}_i \bar{x}_j \neq \bar{y}_{ij} \implies$ branching:
A Branch & Bound algorithm (2)
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
Integer quadratic programming : instance generation

EIQP (Equality Integer Quadratic Problem) : 1 equality constraint.
Integer quadratic programming : instance generation

EIQP (Equality Integer Quadratic Problem) : 1 equality constraint.

IIQP (Inequality Integer Quadratic Problem) : 1 inequality constraint.
Integer quadratic programming : instance generation

EIQP (Equality Integer Quadratic Problem) : 1 equality constraint.

IIQP (Inequality Integer Quadratic Problem) : 1 inequality constraint.

Instance generation :

- Values of $Q$ and $c$ randomly generated in the interval $[-100, 100]$.
- Values of $a_i$ and $d_i$ randomly generated in the interval $[1, 100]$, 
  \[ b = 20 \times \sum_{i=1}^{n} a_i, \quad \text{and} \quad e = 20 \times \sum_{i=1}^{n} d_i. \]
- $\forall i, u_i = 70.$
## Integer quadratic programming: results

**EIQP**: size 40 - 1 hour

|  | $(EIQP_{40})$ |
|---|---|---|---|---|
|  | NC | IQCR | CQCR | B&B |
| ig | 20.3 | **0.1** | 4.2 | 0.1 |
| time (s) | - | 1550 (4) | **877** | 406 (4) |
| nodes | 3510785 | 2589 | 748317 | 635 |
| fg | 13.1 | 0.03 | 0 | 0.3 |

Average SDP time: IQCR: 7000 s, and CQCR: 5 s.

**IIQP**: size 40 - 1 hour

|  | $(IIQP_{40})$ |
|---|---|---|---|---|
|  | NC | IQCR | MIQCR | CQCR |
| ig | 20.8 | **1.2** | **0.1** | 11.4 |
| time (s) | - | 3439 (1) | **713** | 1476(1) |
| nodes | 3990954 | 2017 | 620 | 1969425 |
| fg | 13.7 | 1.0 | 0 | 0.7 |

Average SDP time: IQCR: 5000 s, and CQCR: 5 s.
2 binary quadratic problems

Unconstrained : Pardalos and Rodgers (1990) - 3 hours

<table>
<thead>
<tr>
<th>n densities</th>
<th>UQCR</th>
<th></th>
<th></th>
<th>IQCR</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ig</td>
<td>time (s)</td>
<td>nodes</td>
<td>fg</td>
<td>ig</td>
<td>time (s)</td>
</tr>
<tr>
<td>50_d100</td>
<td>6.6</td>
<td>0.5</td>
<td>1367</td>
<td>0</td>
<td>0.9</td>
<td>2.3</td>
</tr>
<tr>
<td>70_d80</td>
<td>7.3</td>
<td>5.4</td>
<td>10899</td>
<td>0</td>
<td>1.3</td>
<td>17.5</td>
</tr>
<tr>
<td>120_d30</td>
<td>7.1</td>
<td>1263.6</td>
<td>1405507</td>
<td>0</td>
<td>1.5</td>
<td>985</td>
</tr>
<tr>
<td>120_d80</td>
<td>8.7</td>
<td>3905 (6)</td>
<td>2703558</td>
<td>0.5</td>
<td>2.6</td>
<td>2516 (7)</td>
</tr>
</tbody>
</table>

Average SDP time : IQCR : from 1800 s to 20h, and UQCR : about 1200 s

Equality constrained - task allocation problem - 1 hour

<table>
<thead>
<tr>
<th>nbTaches_nbProc</th>
<th>QCR</th>
<th></th>
<th></th>
<th>IQCR</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ig</td>
<td>time (s)</td>
<td>nodes</td>
<td>ig</td>
<td>time (s)</td>
<td>nodes</td>
</tr>
<tr>
<td>20_5</td>
<td>33.2</td>
<td>346.8</td>
<td>521932</td>
<td>7.8</td>
<td>282.5</td>
<td>4859</td>
</tr>
<tr>
<td>25_4</td>
<td>23.9</td>
<td>222.1</td>
<td>313049</td>
<td>8.3</td>
<td>630.2</td>
<td>8686</td>
</tr>
<tr>
<td>25_5</td>
<td>35.0</td>
<td>1875.2 (1)</td>
<td>4118083</td>
<td>11.0</td>
<td>1658.5 (3)</td>
<td>50395</td>
</tr>
<tr>
<td>25_6</td>
<td>45.1</td>
<td>-</td>
<td>3281184</td>
<td>12.9</td>
<td>1454.7 (1)</td>
<td>54872</td>
</tr>
</tbody>
</table>

Average SDP time : IQCR : 3h, and QCR : 25 s
1 Introduction

2 Quadratic convex reformulations

3 A specific Branch and Bound algorithm

4 Computational results

5 Conclusion
Conclusions et results

1. Quadratic convex reformulation to exact solution of integer quadratic programs linearly constrained inspired about 0-1 quadratic programming.
Conclusions et results

1. Quadratic convex reformulation to **exact** solution of **integer quadratic programs** linearly constrained inspired about 0-1 quadratic programming.

2. **Convexifications** can be **efficiently** extended to the **mixed integer** case.
Conclusions et results

1. Quadratic convex reformulation to **exact** solution of **integer quadratic programs** linearly constrained inspired about 0-1 quadratic programming.

2. **Convexifications** can be **efficiently** extended to the **mixed integer** case.

3. for **0-1 programming**, **MIQCR improve** existing convexifications:

Conclusions et results

1. Quadratic convex reformulation to exact solution of integer quadratic programs linearly constrained inspired about 0-1 quadratic programming.

2. Convexifications can be efficiently extended to the mixed integer case.

3. for 0-1 programming, MIQCR improve existing convexifications:
   i) better bound
Conclusions et results

1. Quadratic convex reformulation to exact solution of integer quadratic programs linearly constrained inspired about 0-1 quadratic programming.

2. Convexifications can be efficiently extended to the mixed integer case.

3. for 0-1 programming, MIQCR improve existing convexifications:
   i) better bound
   ii) perturbation of the objective function with inequality constraints
Conclusions et results

1. Quadratic convex reformulation to exact solution of integer quadratic programs linearly constrained inspired about 0-1 quadratic programming.

2. Convexifications can be efficiently extended to the mixed integer case.

3. for 0-1 programming, MIQCR improve existing convexifications :
   i) better bound
   ii) perturbation of the objective function with inequality constraints
   iii) mixed 0-1
Conclusions et results

1. Quadratic convex reformulation to exact solution of integer quadratic programs linearly constrained inspired about 0-1 quadratic programming.

2. Convexifications can be efficiently extended to the mixed integer case.

3. for 0-1 programming, MIQCR improve existing convexifications:
   i) better bound
   ii) pertubation of the objective function with inequality constraints
   iii) mixed 0-1

Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
  - Speed of the SDP solution by mixing both interior point and spectral bundle methods (current work with Franz Rendl).
Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
  - Speed of the SDP solution by mixing both interior point and spectral bundle methods (current work with Franz Rendl).
  - Explore other methods than SDP (exact or heuristic)
Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
  - Speed of the SDP solution by mixing both interior point and spectral bundle methods (current work with Franz Rendl).
  - Explore other methods than SDP (exact or heuristic).

- Multiplicity of optimal values \((\alpha, \beta)\): influence of the choice on the SDP or MIQP solution.
Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
  - Speed of the SDP solution by mixing both interior point and spectral bundle methods (current work with Franz Rendl).
  - Explore other methods than SDP (exact or heuristic).

- Multiplicity of optimal values $(\alpha, \beta)$: influence of the choice on the SDP or MIQP solution.

- Extension of IQCR to mixed integer quadratic programs quadratically constrained.
Outlooks and current work

- Practical improvement of IQCR:
  - Better implementation of the specific Branch & Bound algorithm based on the projection property.
  - Speed of the SDP solution by mixing both interior point and spectral bundle methods (current work with Franz Rendl).
  - Explore other methods than SDP (exact or heuristic).

- Multiplicity of optimal values ($\alpha, \beta$): influence of the choice on the SDP or MIQP solution.

- Extension of IQCR to mixed integer quadratic programs quadratically constrained.

- Test IQCR on more problems.
The 0-1 quadratic knapsack

<table>
<thead>
<tr>
<th></th>
<th>opt</th>
<th>MIQCR</th>
<th>Soutif</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ig</td>
<td>time (s)</td>
</tr>
<tr>
<td>QKP_100_25_1</td>
<td>-24930</td>
<td>0.50</td>
<td>972</td>
</tr>
<tr>
<td>QKP_100_25_2</td>
<td>-18558</td>
<td>0.61</td>
<td>953</td>
</tr>
<tr>
<td>QKP_100_25_3</td>
<td>-35438</td>
<td>0.15</td>
<td>174</td>
</tr>
<tr>
<td>QKP_100_25_4</td>
<td>-20452</td>
<td>0.30</td>
<td>159</td>
</tr>
<tr>
<td>QKP_100_25_5</td>
<td>-36360</td>
<td>0.20</td>
<td>284</td>
</tr>
<tr>
<td>QKP_100_25_6</td>
<td>-61494</td>
<td>0.22</td>
<td>189</td>
</tr>
<tr>
<td>QKP_100_25_7</td>
<td>-50382</td>
<td>0.08</td>
<td>112</td>
</tr>
<tr>
<td>QKP_100_25_8</td>
<td>-3752</td>
<td>1.40</td>
<td>69</td>
</tr>
<tr>
<td>QKP_100_25_9</td>
<td>-56525</td>
<td>0.10</td>
<td>214</td>
</tr>
<tr>
<td>average</td>
<td></td>
<td>0.39</td>
<td>347</td>
</tr>
</tbody>
</table>
random generated mixed integer problems

<table>
<thead>
<tr>
<th>Name</th>
<th>ig</th>
<th>CSDP time(s)</th>
<th>Cplex time (s)</th>
<th>nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQP1_13_27_1</td>
<td>0</td>
<td>1240</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_13_27_2</td>
<td>0</td>
<td>1252</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_13_27_3</td>
<td>0</td>
<td>1587</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_13_27_4</td>
<td>0</td>
<td>1294</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_13_27_5</td>
<td>0</td>
<td>1246</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>average</td>
<td>0</td>
<td>1323.8</td>
<td>4.2</td>
<td>1.2</td>
</tr>
<tr>
<td>MQP1_16_34_1</td>
<td>0</td>
<td>6088</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_16_34_2</td>
<td>1.07</td>
<td>4746</td>
<td>57</td>
<td>58</td>
</tr>
<tr>
<td>MQP1_16_34_3</td>
<td>0</td>
<td>4747</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>MQP1_16_34_4</td>
<td>0</td>
<td>4726</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_16_34_5</td>
<td>0.15</td>
<td>4744</td>
<td>42</td>
<td>51</td>
</tr>
<tr>
<td>average</td>
<td>0.24</td>
<td>5010.2</td>
<td>27.4</td>
<td>23.2</td>
</tr>
<tr>
<td>MQP1_20_40_1</td>
<td>0</td>
<td>14756</td>
<td>55</td>
<td>9</td>
</tr>
<tr>
<td>MQP1_20_40_2</td>
<td>0.20</td>
<td>17959</td>
<td>142</td>
<td>52</td>
</tr>
<tr>
<td>MQP1_20_40_3</td>
<td>0.37</td>
<td>14530</td>
<td>46</td>
<td>14</td>
</tr>
<tr>
<td>MQP1_20_40_4</td>
<td>0</td>
<td>14682</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>MQP1_20_40_5</td>
<td>0.15</td>
<td>15835</td>
<td>61</td>
<td>16</td>
</tr>
<tr>
<td>average</td>
<td>0.14</td>
<td>15552.4</td>
<td>68.4</td>
<td>16.2</td>
</tr>
</tbody>
</table>
Designing an optimal portfolio with mean variance model (1)

\[
(\text{QP}^{1}_{\text{inv}}) \quad \begin{cases} 
\text{Min} & f(x) = x^T Q x \\
\text{s.t.} & \sum_{i=1}^{n} x_i = 1 \\
& \sum_{i=1}^{n} R_i x_i \geq \rho \\
& x_i \leq y_i \quad i = 1, \ldots, n \\
& \sum_{i=1}^{n} y_i \leq N \\
& y_i \in \{0, 1\}^n \quad i = 1, \ldots, n 
\end{cases}
\]
Application to an example

\[
\begin{aligned}
\left\{ \begin{array}{l}
\text{Min } f(x) = x^T \begin{pmatrix}
21.5 & -17.25 & -2.75 & 2.75 & -14.75 \\
-17.25 & 56.5 & -0.25 & 0.25 & -3.5 \\
-2.75 & -0.25 & 0.5 & -0.5 & 3.25 \\
2.75 & 0.25 & -0.5 & 0.5 & -3.25 \\
-14.75 & -3.5 & 3.25 & -3.25 & 49.5
\end{pmatrix} x \\
\text{s.t. } 4x_1 + 2x_2 + 3x_3 + 3x_4 + 4x_5 \geq 3.3 \\
x_i \leq y_i \\
y_1 + y_2 + y_3 + y_4 + y_5 \leq 3 \\
y_i \in \{0, 1\}^n
\end{array} \right. \\
\end{aligned}
\]

Direct cplex continuous relaxation value : 0.478
Cplex continuous relaxation value after reformulation by MIQCR : 0.581
Optimal integer solution value : 0.581
Designing an optimal portfolio with mean variance model (2)

\[
\begin{align*}
\text{(QP}_{\text{inv}}^2) \quad \begin{cases} 
\text{Min} & f(x) = x^T Q x \\
\text{s.t.} & \sum_{i=1}^{n} x_i = 1 \\
& \sum_{i=1}^{n} R_i x_i \geq \rho \\
& x_i \leq l_i y_i \quad i = 1, \ldots, n \\
& x_i \geq u_i y_i \quad i = 1, \ldots, n \\
& y_i \in \{0, 1\}^n \quad i = 1, \ldots, n
\end{cases}
\end{align*}
\]
Application to an example

\[
(QP_{inv}^2) \left\{ \begin{array}{l}
\text{Min} \quad f(x) = x^T \begin{pmatrix}
21.5 & -17.25 & -2.75 & 2.75 & -14.75 \\
-17.25 & 56.5 & -0.25 & 0.25 & -3.5 \\
-2.75 & -0.25 & 0.5 & -0.5 & 3.25 \\
2.75 & 0.25 & -0.5 & 0.5 & -3.25 \\
-14.75 & -3.5 & 3.25 & -3.25 & 49.5
\end{pmatrix} x \\
\text{s.t.} \quad 4x_1 + 2x_2 + 3x_3 + 3x_4 + 4x_5 \geq 3.3 \\
x_i \leq 0.5y_i \quad \text{for} \quad i = 1, \ldots, 5 \\
x_i \geq 0.2y_i \quad \text{for} \quad i = 1, \ldots, 5 \\
y_i \in \{0, 1\}^n
\end{array} \right.
\]

Direct cplex continuous relaxation value : 0.551
Cplex continuous relaxation value after reformulation by MIQCR : 1.481
Optimal integer solution value : 1.485