Exact quadratic convex reformulations of mixed-integer quadratically constrained problems

Alain Billionnet, Sourour Elloumi, Amélie Lambert

Exact quadratic convex reformulations of Mixed-Integer Quadratically Constrained Problems

Alain Billionnet[1], Sourour Elloumi[1], and Amélie Lambert[2]

1.CEDRIC-ENSIIE, 1 square de la résistance, 91025 Evry cedex, France
2. CEDRIC-Cnam, 292 rue Saint-Martin, F-75141 Paris cedex 03, France

Abstract. We propose a solution approach for the general problem (QP) of minimizing a quadratic function of bounded integer variables subject to a set of quadratic constraints. The resolution is based on the reformulation of the original problem (QP) into an equivalent quadratic problem whose continuous relaxation is convex, so that it can be effectively solved by a branch-and-bound algorithm based on quadratic convex relaxation. We concentrate our efforts on finding a reformulation such that the continuous relaxation bound of the reformulated problem is as tight as possible.
Furthermore, we extend our method to the case of mixed-integer quadratic problems with the following restriction: all quadratic sub-functions of purely continuous variables are already convex.
Finally, we illustrate the different results of the article by small examples and we present some computational experiments on pure-integer and mixed-integer instances of (QP). Most of the considered instances with up to 53 variables can be solved by our approach combined with the use of Cplex.

Key words: Integer quadratic programming, Equivalent convex reformulation, semidefinite programming, branch-and-bound algorithm

1 Introduction

In this paper, we aim at the exact solution of mixed-integer quadratically constrained programs. For this, we first consider the pure integer case, i.e. the following program (QP):

$$(QP)\begin{cases} \min f_0(x) & \\ \text{s.t.} & \\ \quad f_r(x) \leq b_r & r = 1, \ldots, m \\ \quad \ell_i \leq x_i \leq u_i & i \in I \\ \quad x_i \in \mathbb{N} & i \in I \end{cases}$$

where

$$f_r(x) = \langle Q_r, xx^T \rangle + c_r^T x \quad \forall\ r = 0, \ldots, m$$

with $\langle A, B \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij}$, and for $r = 0, \ldots, m$, Q_r is a symmetric $n \times n$ matrix, $c_r \in \mathbb{R}^n$, $b_r \in \mathbb{R}$, $I = \{1, \ldots, n\}$. Variables x_i are integers and are bounded by $u_i \in \mathbb{N}$ and $\ell_i \in \mathbb{N}$. Without loss of generality, we shall suppose $\ell_i = 0$ since it is possible to change variable x_i into $x_i - \ell_i$. We assume the feasible domain of (QP) to be non-empty.

This general program trivially contains the case where there are quadratic equalities, since an equality can be replaced by two inequalities. It also contains the case of linear constraints since a linear equality is a quadratic constraint with a zero quadratic part.

Program (QP) belongs to the class of $\mathcal{NP}$-hard [13] Mixed-Integer Non Linear Programs (MINLP). General approaches to solve MINLP are based on global optimization techniques [5,11,12,17,24,25]. More specific methods are also available to solve quadratically constrained programs, where all variables are continuous [2, 4, 18, 22]. These approaches can handle binary quadratic programming using identity $x_i^2 = x_i$. Hence, they are also able to handle (QP) by applying a binary expansion of each integer variable. This method will have to consider the large size of the equivalent binary quadratic program.

Other methods are available to obtain convex relaxations of mixed-integer quadratically constrained programs. A general technique for computing such relaxations is semi-definite programming [3]. A recent paper [23] proposes several ideas for keeping the tightness of these relaxations while improving their computation times.

References [20] and [21] provide effective approaches for solving mixed-integer quadratic problems with real variables and 0-1 variables. The considered problem consists in optimizing a quadratic function subject to quadratic constraints, but the quadratic parts of the objective function and constraints include only continuous variables. In contrast, the linear parts include both continuous variables and 0-1 variables.

We introduced in [8] the Mixed Integer Quadratic Convex Reformulation (`MIQCR`) approach. This method handles mixed-integer quadratic problems with linear equality constraints only. The idea of `MIQCR` is to design the tightest possible equivalent program to (QP) with a convex objective function, within a convex reformulation scheme. This best equivalent problem can be computed using the dual solution of a semidefinite relaxation of the initial problem, and further solved by a branch-and-bound algorithm based on quadratic convex relaxation.

In this paper, we present a method to solve (QP) based on the reformulation of the original problem into an equivalent quadratic problem whose continuous relaxation is convex. This new method handles quadratic constraints. At this

aim, we consider a different reformulation scheme than in `MIQCR`. This scheme allows further extension to the case where some of the variables are continuous.

The structure of the paper is as follows. In Section 2, we introduce a new family of equivalent formulations to (QP). Each of these equivalent formulations has a convex continuous relaxation.

In Section 3, we focus on finding the tightest equivalent formulation within this family. We show that this best equivalent formulation can be deduced from a semidefinite relaxation to (QP), and that, surprisingly, it consists in linearizing all the constraints.

In Section 4, we study the equality constrained case. After a brief recall of the `MIQCR` method, we highlight its links with our current work. Moreover, we prove another interesting result: for equality constraints, any linear or quadratic convex reformulation of these constraints can be used to build the best reformulation.

In Section 5, we show how the whole framework can be extended to the case of mixed-integer variables, with the restriction that the quadratic sub-functions of purely continuous variables are already convex.

Throughout this paper, we present some small numerical examples to illustrate different aspects of the method. In Section 6, we present computational results on various instances of (QP) to show the effectiveness of the approach. Section 7 draws a conclusion.

Notation
The following notation is used throughout the paper. We denote by $v(P)$ the optimal value of a mathematical program (P). For a vector $u \in \mathbb{R}^n$, $diag(u)$ denotes a diagonal matrix whose i^{th} diagonal element equals u_i. We denote by $\mathcal{S}_n$ the set of $n \times n$ symmetric matrices, by $\mathcal{S}_n^+$ the set of positive semidefinite matrices of $\mathcal{S}_n$ and $M \succeq 0$ means that $M \in \mathcal{S}_n^+$. We also denote by $\mathbf{0}_n$ the zero $n \times n$ matrix and by I^2 the Cartesian product of a set I by itself.

2 A general family of convex equivalent formulations to (QP)

In this section, we introduce a family of equivalent formulations to (QP). These formulations use additional variables y_{ij} that are enforced, by linear constraints, to be equal to the product $x_i x_j$. Any quadratic function is then formulated as a sum of a quadratic function of the x variables and a linear function of the y variables.

Equivalent formulation of the quadratic functions
We first introduce n^2 new variables y that will satisfy $y_{ij} = x_i x_j \quad \forall (i,j) \in I^2$, or, equivalently $Y = xx^T$.

Then, for $r = 0, \ldots, m$ we consider a positive semidefinite matrix $S_r \in \mathcal{S}_n^+$ and replace the quadratic function $f_r(x)$ by a convex function $f_{r,S_r}(x,Y)$ that is

equal to f_r under the condition $Y = xx^T$. We define function $f_{r,S_r}(x,Y)$ as follows :

$$f_{r,S_r}(x,Y) = \langle S_r, xx^T\rangle + c_r^T x + \langle Q_r - S_r, Y\rangle \qquad \forall\ r = 0,\ldots,m$$

Hence, problem (QP) is equivalently stated as:

$$\begin{cases} \min \quad \langle S_0, xx^T\rangle + c_0^T x + \langle Q_0 - S_0, Y\rangle & & \\ \text{s.t.} & & \\ \quad \langle S_r, xx^T\rangle + c_r^T x + \langle Q_r - S_r, Y\rangle \le b_r & r = 1,\ldots,m & \\ \quad 0 \le x_i \le u_i & i \in I & (1)\\ \quad x_i \in \mathbb{N} & i \in I & (2)\\ \quad y_{ij} = x_i x_j & (i,j) \in I^2 & (3) \end{cases}$$

Because matrices S_r are positive semidefinite, functions $f_{r,S_r}(x,Y)$ are convex.

Linearization of the quadratic constraints $Y = xx^T$ [7]
We now concentrate our effort on replacing Constraints (3) together with (1) and (2) by a set of linear inequalities. For this, each variable x_i is replaced by its unique binary decomposition $x_i = \sum_{k=0}^{\lfloor \log(u_i)\rfloor} 2^k t_{ik}$, we can then express the product y_{ij} of two variables x_i and x_j as a linear function of the products of a variable x by a variable t: $y_{ij} = \sum_{k=0}^{\lfloor \log(u_i)\rfloor} x_j 2^k t_{ik}$. These products are then linearized by replacing them with a variable z and adding the appropriate linear constraints. To get closer to the convex hull, we furthermore add the McCormick inequalities [19]. We obtain the following set P_{xYzt}:

$$P_{xYzt}\begin{cases} x_i = \displaystyle\sum_{k=0}^{\lfloor \log(u_i)\rfloor} 2^k t_{ik} & i \in I \\ y_{ij} = \displaystyle\sum_{k=0}^{\lfloor \log(u_i)\rfloor} 2^k z_{ijk} & (i,j) \in I^2 \\ z_{ijk} \le u_j t_{ik} & (i,k) \in E,\ j \in I \\ z_{ijk} \le x_j & (i,k) \in E,\ j \in I \\ z_{ijk} \ge x_j - u_j(1 - t_{ik}) & (i,k) \in E,\ j \in I \\ z_{ijk} \ge 0 & (i,k) \in E,\ j \in I \\ t_{ik} \in \{0,1\} & (i,k) \in E \\ y_{ii} \ge x_i & i \in I \\ y_{ij} = y_{ji} & (i,j) \in I^2,\ i < j \\ y_{ij} \ge u_j x_i + u_i x_j - u_i u_j & (i,j) \in I^2,\ i \le j \\ y_{ij} \ge 0 & (i,j) \in I^2,\ i \le j \end{cases}$$

where $E = \{(i,k) : i = 1,\dots,n,\ k = 0,\dots \lfloor \log(u_i) \rfloor\}$. The number of binary variables is $N = |E| = n + \sum_{i=1}^{n} (\lfloor \log(u_i) \rfloor)$ and the number of real variables is $n + n^2 + nN$, so that the set P_{xYzt} has $O(nN)$ variables and constraints.

The family of integer convex quadratic equivalent formulations to (QP)
To sum up, for any set of positive semidefinite matrices S_r $(r = 0,\dots,m)$, we replace Constraints (1)-(3) by the set P_{xYzt}. We obtain the following equivalent problem to (QP):

$$(QP_{S_0,\dots,S_m}) \begin{cases} \min \langle S_0, xx^T \rangle + c_0^T x + \langle Q_0 - S_0, Y \rangle \\ \text{s.t.} \\ \quad \langle S_r, xx^T \rangle + c_r^T x + \langle Q_r - S_r, Y \rangle \leq b_r \quad r = 1,\dots,m \\ \quad (x, Y, z, t) \in P_{xYzt} \end{cases}$$

Hence, we built an infinite family of equivalent problems to (QP).

The advantage of $(QP_{S_0,\dots,S_m})$ is that the only non-convexity comes from the integrality constraints $t_{ik} \in \{0,1\}$. Relaxing these constraints leads to a quadratic convex optimization problem which optimal value can be computed in polynomial time. Hence, problem $(QP_{S_0,\dots,S_m})$ can for example be handled by a mixed-integer quadratic programming solver which performs a branch-and-cut algorithm to solve it.

This general family of reformulations includes two extreme cases. The first one is $S_r = Q_r$, when all Q_r matrices are already positive semidefinite. In that case, functions $f_r(x)$ are left unchanged. The second one is when all S_r are zero-matrices. In this case, the reformulation consists in replacing each product of two x variables by a Y variable. This amounts to a complete linearization [1], a classical approach in discrete quadratic optimization.

3 Computing the best convex equivalent formulation

In this section, we will focus on finding the positive semidefinite matrices $S_0^*,\dots,S_m^*$ such that the continuous relaxation bound of problem $(QP_{S_0,\dots,S_m})$ is as tight as possible.

Let us first recall a result already proved in [8] (Theorem 1). In P_{xYzt}, replace the integrality constraints $t_{ik} \in \{0,1\}$ by $t_{ik} \in [0,1]$ and project the obtained polyhedron on variables x and Y. The projected polyhedron is the following

$\overline{P}_{xY}$:

$$\overline{P}_{xY}\begin{cases} y_{ii} \geq x_i & i \in I & (4)\\ y_{ij} = y_{ji} & (i,j) \in I^2,\ i < j & (5)\\ y_{ij} \geq u_j x_i + u_i x_j - u_i u_j & (i,j) \in I^2,\ i \leq j & (6)\\ y_{ij} \geq 0 & (i,j) \in I^2,\ i \leq j & (7)\\ y_{ij} \leq u_i x_j & (i,j) \in I^2,\ i \leq j & (8)\\ y_{ij} \leq u_j x_i & (i,j) \in I^2,\ i \leq j & (9) \end{cases}$$

Hence, since variables z and t are not in the objective function, the continuous relaxation optimal value of $(QP_{S_0,\ldots,S_m})$ can be computed by solving the following smaller problem $(RQP_{S_0,\ldots,S_m})$ with x and Y variables only:

$$(RQP_{S_0,\ldots,S_m})\begin{cases} \min\langle S_0, xx^T\rangle + c_0^T x + \langle Q_0 - S_0, Y\rangle & \\ \text{s.t.} & \\ \quad \langle S_r, xx^T\rangle + c_r^T x + \langle Q_r - S_r, Y\rangle \leq b_r \quad r = 1,\ldots,m & (10)\\ \quad (x,Y) \in \overline{P}_{xY} & (11) \end{cases}$$

We want to find positive semidefinite matrices $S_0^*, \ldots, S_m^*$ such that the continuous relaxation value of $(QP_{S_0^*,\ldots,S_m^*})$ is maximized. This amounts to solving the following problem (CP):

$$(CP)\begin{cases} \max\limits_{S_0,\ldots,S_m \succeq 0} \{v(RQP_{S_0,\ldots,S_m})\} \end{cases}$$

3.1 Finding an optimal solution to (CP)

Here, we prove that an optimal solution to (CP) can be deduced from a dual solution of the following program (SDP) which is also a semidefinite relaxation of (QP). More precisely, to build (SDP) we perform a classical semidefinite relaxation of (QP), and we add McCormick inequalities (13)–(16) [19], and inequalities (17) that are satisfied since $x_i \in \mathbb{N}$.

$$
(SDP)\begin{cases}
\min f(X,x) = \langle Q_0, X\rangle + c_0^T x & \\
\text{s.t.} & \\
\quad \langle Q_r, X\rangle + c_r^T x \le b_r & r = \{1,\dots,m\} \quad (12)\\
\quad X_{ij} - u_j x_i \le 0 & (i,j)\in I^2,\ i\le j \quad (13)\\
\quad X_{ij} - u_i x_j \le 0 & (i,j)\in I^2,\ i\le j \quad (14)\\
\quad -X_{ij} + u_j x_i + u_i x_j \le u_i u_j & (i,j)\in I^2,\ i\le j \quad (15)\\
\quad -X_{ij} \le 0 & (i,j)\in I^2,\ i\le j \quad (16)\\
\quad -X_{ii} + x_i \le 0 & i\in I \quad (17)\\
\quad \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 & \quad (18)\\
\quad x\in\mathbb{R}^n \quad X\in\mathcal{S}_n &
\end{cases}
$$

Theorem 1. *The optimal value of* (CP) *is equal to the optimal value of* (SDP).

Proof.
$\diamond$ Let us firstly prove that $v(CP) \le v(SDP)$
Let $\bar{S}_0,\dots,\bar{S}_m \in \mathcal{S}_n^+$ be any feasible solution to (CP). To prove that $v(CP) \le v(SDP)$ we prove that from any feasible solution $(\bar{X},\bar{x})$ to (SDP), we can build a feasible solution (x,Y) to $(RQP_{\bar{S}_0,\dots,\bar{S}_m})$ with a lower objective value, i.e., satisfying $\langle \bar{S}_0, \bar{x}\bar{x}^T\rangle + c_0^T\bar{x} + \langle Q_0 - \bar{S}_0, \bar{X}\rangle \le f(\bar{X},\bar{x})$.

i) Take $x = \bar{x}$. Variables Y can be seen as a symmetric matrix, so we can take $y_{ij} = \bar{X}_{ij}$, $\forall (i,j)\in I^2$. We prove that this solution (x,Y) is feasible to $(RQP_{\bar{S}_0,\dots,\bar{S}_m})$. Constraints(4)-(9) are obviously satisfied. We now prove that Constraints (10) are satisfied. We have:

$$
\begin{aligned}
\langle \bar{S}_r, xx^T\rangle + c_r^T x + \langle Q_r - \bar{S}_r, Y\rangle &= \langle \bar{S}_r, \bar{x}\bar{x}^T\rangle + c_r^T\bar{x} + \langle Q_r - \bar{S}_r, \bar{X}\rangle \\
&= \langle \bar{S}_r, \bar{x}\bar{x}^T - \bar{X}\rangle + c_r^T\bar{x} + \langle Q_r, \bar{X}\rangle \\
&\le b_r \qquad \text{from Constraints (12), since} \\
&\qquad\qquad \bar{S}_r \succeq 0, \text{ and } \bar{x}\bar{x}^T - \bar{X} \preceq 0.
\end{aligned}
$$

ii) Prove that $\langle \bar{S}_0, \bar{x}\bar{x}^T\rangle + c_0^T\bar{x} + \langle Q_0 - \bar{S}_0, \bar{X}\rangle - \langle Q_0, \bar{X}\rangle - c_0^T\bar{x} \le 0$ or that $\langle \bar{S}_0, \bar{x}\bar{x}^T - \bar{X}\rangle \le 0$. This last inequality follows from $\bar{S}_0 \succeq 0$ and $\bar{x}\bar{x}^T - \bar{X} \preceq 0$.

$\diamond$ Let us secondly prove that $v(CP) \ge v(SDP)$ or equivalently $v(CP) \ge v(DSDP)$ where $(DSDP)$ is the dual of (SDP). The following problem $(DSDP)$ is the dual of (SDP):

$$(DSDP)\begin{cases} \max g(\alpha,\Phi) = -\sum_{r=1}^{m}\alpha_r b_r - \langle \Phi^3, uu^T\rangle \\ \text{s.t.} \\ \quad Q_0 + \sum_{r=1}^{m}\alpha_r Q_r + \Phi \succeq 0 & (19) \\ \quad c_0 + \sum_{r=1}^{m}\alpha_r c_r - (\Phi^1+\Phi^2-2\Phi^3)^T u + \varphi \geq 0 & (20) \\ \quad \Phi = \Phi^1+\Phi^2-\Phi^3-\Phi^4 - diag(\varphi) & (21) \\ \quad \alpha \in \mathbb{R}^m_+,\ \Phi \in \mathcal{S}_n,\ \Phi^1,\Phi^2,\Phi^3,\Phi^4 \in \mathcal{S}^+_n,\ \varphi \in \mathbb{R}^n_+ \end{cases}$$

where $\alpha \in \mathbb{R}^m_+$ are the dual variables associated to constraints (12), and Φ^i, $i = 1,\ldots,4$ are the positive semidefinite matrices built from the dual variables θ associated with constraints (13), (14), (15), (16), respectively. For instance, if θ^1_{ij} is the dual variable associated to constraint (13), then $\Phi^1_{ij} = \Phi^1_{ji} = \frac{\theta^1_{ij}}{2}$ for $i < j$, and $\Phi^1_{ii} = \theta^1_{ii}$. φ are the dual variables associated to constraints (17). As mentioned in Constraint (21) we have $\Phi = \Phi^1+\Phi^2-\Phi^3-\Phi^4-diag(\varphi)$.

Let $(\bar\alpha, \bar\Phi^1, \bar\Phi^2, \bar\Phi^3, \bar\Phi^4, \bar\varphi)$ be a feasible solution to $(DSDP)$ and, by Constraint (21), let $\bar\Phi = \bar\Phi^1+\bar\Phi^2-\bar\Phi^3-\bar\Phi^4-diag(\bar\varphi)$, then we build the following positive semidefinite matrices:

$$\bar S_r = \mathbf{0}_n \qquad r = 1,\ldots,m$$
$$\bar S_0 = Q_0 + \sum_{r=1}^{m}\bar\alpha_r Q_r + \bar\Phi$$

$(\bar S_0,\ldots,\bar S_m)$ form a feasible solution to (CP). The objective value of this solution is equal to $v(RQP_{\bar S_0,\ldots,\bar S_m})$.

We now prove that $v(RQP_{\bar S_0,\ldots,\bar S_m}) \geq v(DSDP)$. For this, we prove that for any feasible solution $(\bar x, \bar Y)$ to $(RQP_{\bar S_0,\ldots,\bar S_m})$, the associated objective value is not smaller then $g(\bar\alpha,\bar\Phi)$. Denote by Δ the difference between the objective values, i.e., $\Delta = \langle \bar S_0, \bar x\bar x^T\rangle + c_0^T\bar x + \langle Q_0 - \bar S_0, \bar Y\rangle - g(\bar\alpha,\bar\Phi)$. We below prove that $\Delta \geq 0$

$$\begin{aligned}
\Delta &= \langle \bar{S}_0, \bar{x}\bar{x}^T \rangle + c_0^T \bar{x} + \langle Q_0 - \bar{S}_0, \bar{Y} \rangle + \sum_{r=1}^{m} \bar{\alpha}_r b_r + \langle \bar{\Phi}^3, uu^T \rangle \\
&\geq c_0^T \bar{x} - \langle \sum_{r=1}^{m} \bar{\alpha}_r Q_r + \bar{\Phi}, \bar{Y} \rangle + \sum_{r=1}^{m} \bar{\alpha}_r b_r + \langle \bar{\Phi}^3, uu^T \rangle \quad \text{since } \bar{S}_0 \succeq 0 \\
&= c_0^T \bar{x} + \sum_{r=1}^{m} \bar{\alpha}_r (b_r - \langle Q_r, \bar{Y} \rangle) - \langle \bar{\Phi}, \bar{Y} \rangle + \langle \bar{\Phi}^3, uu^T \rangle \\
&\geq c_0^T \bar{x} + \sum_{r=1}^{m} \bar{\alpha}_r c_r^T \bar{x} - \langle \bar{\Phi}, \bar{Y} \rangle + \langle \bar{\Phi}^3, uu^T \rangle
\end{aligned}$$

as $c_r^T \bar{x} + \langle Q_r, \bar{Y} \rangle \leq b_r$ and $\bar{\alpha}_r \geq 0$. Moreover, by Constraint (21) we get:

$$\begin{aligned}
\Delta &\geq c_0^T \bar{x} + \sum_{r=1}^{m} \bar{\alpha}_r c_r^T \bar{x} - \langle \bar{\Phi}^1 + \bar{\Phi}^2 - \bar{\Phi}^3 - \bar{\Phi}^4 - diag(\bar{\varphi}), \bar{Y} \rangle + \langle \bar{\Phi}^3, uu^T \rangle \\
&\geq c_0^T \bar{x} + \sum_{r=1}^{m} \bar{\alpha}_r c_r^T \bar{x} - \langle \bar{\Phi}^1, \bar{Y} \rangle - \langle \bar{\Phi}^2, \bar{Y} \rangle + \langle \bar{\Phi}^3, \bar{Y} + uu^T \rangle + \langle \bar{\Phi}^4, \bar{Y} \rangle + \langle diag(\bar{\varphi}), \bar{Y} \rangle
\end{aligned}$$

By Constraints (4)–(9), and since all the coefficients of $\bar{\Phi}^1, \bar{\Phi}^2, \bar{\Phi}^3, \bar{\Phi}^4$, and $\bar{\varphi}$ are non-negative, we get:

$$\begin{aligned}
\Delta &\geq c_0^T \bar{x} + \sum_{r=1}^{m} \bar{\alpha}_r c_r^T \bar{x} - \langle \bar{\Phi}^1, \bar{x}^T u \rangle - \langle \bar{\Phi}^2, \bar{x}^T u \rangle + \langle \bar{\Phi}^3, 2\bar{x}^T u \rangle + \bar{\varphi}^T \bar{x} \\
&\geq \Big(c_0 + \sum_{r=1}^{m} \bar{\alpha}_r c_r - (\bar{\Phi}^1 + \bar{\Phi}^2 - 2\bar{\Phi}^3)^T u + \bar{\varphi} \Big)^T \bar{x} \\
&\geq 0 \qquad \text{since } \bar{x} \geq 0 \text{ and by Constraint (20).}
\end{aligned}$$

□

From the proof of Theorem 1, we can characterize an optimal solution to (CP).

Corollary 1. *An optimal solution S_r^*, $r = 0, ..., m$ to (CP) amounts to linearizing the constraints by taking*

$$S_r^* = \mathbf{0}_n \text{ for } r = 1, ..., m$$

and, for the objective function, we take

$$S_0^* = Q_0 + \sum_{r=1}^{m} \alpha_r^* Q_r + \Phi^*$$

where α_r^* and Φ^* are computed from an optimal solution of the dual $(DSDP)$ of the semidefinite programming problem (SDP):

i) α^* *is the vector of optimal dual variables associated with constraints (12) of* (SDP),

ii) Φ^*_{ij} *are computed as* $\Phi^*_{ij} = \Phi^{1*}_{ij} + \Phi^{2*}_{ij} - \Phi^{3*}_{ij} - \Phi^{4*}_{ij}$, *for* $i \neq j$, *and* $\Phi^*_{ii} = \Phi^{1*}_{ii} + \Phi^{2*}_{ii} - \Phi^{3*}_{ii} - \Phi^{4*}_{ii} - \Phi^{5*}_{i}$, *where* Φ^{1*}, Φ^{2*}, Φ^{3*}, Φ^{4*} *are the symmetric matrices built from the optimal dual variables associated with constraints (13), (14), (15), (16), respectively, and* Φ^{5*} *the vector of dual variables associated with constraints (17).*

To sum up, let us write the optimal equivalent formulation:

$$(QP^*) = (QP_{S_0^*,\mathbf{o}_n,\ldots,\mathbf{o}_n}) \begin{cases} \min\langle Q_0 + \sum_{r=1}^{m} \alpha_r^* Q_r + \Phi^*, xx^T\rangle + c_0^T x - \langle \sum_{r=1}^{m} \alpha_r^* Q_r + \Phi^*, Y\rangle \\ \text{s.t.} \\ \quad c_r^T x + \langle Q_r, Y\rangle \leq b_r \quad r = 1,\ldots,m \\ \quad (x, Y, z, t) \in P_{xYzt} \end{cases} \tag{22}$$

A discussion on the optimal equivalent formulation
An important fact is that the continuous relaxation value of program (QP^*) is equal to the optimal value of the semi-definite relaxation (SDP). Hence, in a branch-and-bound algorithm for solving (QP^*), the root-node gap is the same as the SDP-relaxation gap which is known to be strong. In [6], quadratic programs with continuous bounded variables are considered and this SDP relaxation, without Constraints (17) which is specific to the integer variables case, is called "Shor plus RLT" relaxation. It is recalled in [6] that this relaxation dominates six other considered SDP relaxations. It is further shown that it provides the same bound as the doubly nonnegative relaxation.

In presence of a linear inequality $a^T x \leq b$, we consider it as a quadratic constraint with a zero quadratic part. Moreover, in order to improve the tightness of the relaxation, we add the following quadratic inequality $x^T a a^T x \leq b^2$.

3.2 A solution algorithm for (QP)

From Theorem 1 and Corollary 1, we can deduce Algorithm 1 to solve (QP).

Algorithm 1 Solution algorithm to (QP)

step 1: Solve the semidefinite program (SDP).

step 2: Deduce $(S_0^*, \ldots, S_m^*)$ as described in Corollary 1.

step 3: Solve the program $(QP_{S_0^*,\ldots,S_m^*})$ by a MIQP solver.
(Its continuous relaxation is a convex program with an optimal value equal to the optimal value of (SDP)*)*

3.3 Illustrative examples

In this section, we first examine the solution of an example by Algorithm 1. Then, we show on a convex example that Algorithm 1 can lead to a tighter continuous relaxation bound than the direct solution by Cplex [**?**].

Illustration of Algorithm 1

Consider the following example whose optimal value is equal to -1872 for $x^* = (9, 0, 20, 14)$:

$$(Ex)\begin{cases}\min\ f(x) = -4x_1x_2 - 4x_1x_4 + 6x_2x_4 - 3x_3^2 - 2x_3x_4 + 2x_4^2 \\ \text{s.t.} \\ \quad 8x_1^2 + 5x_2^2 + 8x_2x_3 + 4x_2x_4 + 2x_4^2 \leq 1080 \\ \quad 0 \leq x_1 \leq 11 \\ \quad 0 \leq x_2 \leq 14 \\ \quad 0 \leq x_3 \leq 20 \\ \quad 0 \leq x_4 \leq 16 \\ \quad x_i \in \mathbb{N} \qquad\qquad i = \{1,2,3,4\}\end{cases}$$

Following Algorithm 1, we first solve the semidefinite relaxation (SDP_{Ex}):

$$(SDP_{Ex})\begin{cases}\min\ f(X,x) = -4X_{12} - 4X_{14} + 6X_{24} - 3X_{33} - 2X_{34} + 2X_{44} \\ \text{s.t.}\ \ 8X_{11} + 5X_{22} + 8X_{23} + 4X_{24} + 2X_{44} \leq 1080 \\ \quad X_{ij} \leq u_j x_i \\ \quad X_{ij} \leq u_i x_j \\ \quad -X_{ij} \leq -u_j x_i - u_i x_j + u_i u_j \\ \quad -X_{ij} \leq 0 \\ \quad -X_{ii} \leq -x_i \\ \quad \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ \quad x \in \mathbb{R}^4 \quad X \in \mathbf{S}_4\end{cases}$$

We obtain the following optimal dual solution:

- $\alpha_1^* = 0.3773$
- $\Phi^* = \begin{pmatrix} 0 & 0.31 & -0.18 & 0 \\ 0.31 & 1.32 & -1.00 & -1.36 \\ -0.18 & -1.00 & 3.61 & 0.89 \\ 0 & -1.36 & 0.89 & 0 \end{pmatrix}$

We build matrix $S_0^* = Q_0 + \alpha_1^* Q_1 + \Phi^*$, and we get:

$$S_0^* = \begin{pmatrix} 3.02 & -1.69 & -0.18 & -2 \\ -1.69 & 3.21 & 0.51 & 2.39 \\ -0.18 & 0.51 & 0.61 & -0.11 \\ -2 & 2.39 & -0.11 & 2.75 \end{pmatrix} \text{ and then } Q_0 - S_0^* = \begin{pmatrix} -3.02 & -0.31 & 0.18 & 0 \\ -0.31 & -3.21 & -0.51 & 0.61 \\ 0.18 & -0.51 & -3.61 & -0.89 \\ 0 & 0.61 & -0.89 & -0.75 \end{pmatrix}$$

Then, to solve (Ex), we reformulate it into the following quadratic program with linear constraints (Ex^*)

$$(Ex^*)\begin{cases}\min\ f_{0,S_0^*}(x,Y) = -3.02y_{11} - 0.62y_{12} + 0.36y_{13} - 3.21y_{22} - 1.02y_{23} \\ \qquad +1.22y_{24} - 3.61y_{33} - 1.78y_{34} - 0.75y_{44} + 3.02x_1^2 - 3.38x_1x_2 - 0.36x_1x_3 \\ \qquad -4x_1x_4 + 3.21x_2^2 + 1.02x_2x_3 + 4.78x_2x_4 + 0.61x_3^2 - 0.22x_3x_4 + 2.75x_4^2 \\ \text{s.t.}\ \ 8y_{11} + 5y_{22} + 8y_{23} + 4y_{24} + 2y_{44} \leq 1080 \\ \quad (x,Y,z,t) \in P_{xYzt}\end{cases}$$

The continuous relaxation value of (Ex^*) is equal to -1887.32, the initial gap is thus of 0.82%. The continuous relaxation value of the complete linearization

(i.e. $S_0 = \mathbf{0}_n$) of (Ex) is equal to -2148.83 leading to a gap of 14.79%.

Improving the continuous relaxation bound for already convex instances

Through this instance, we illustrate the interest of using our algorithm for convex instances. We consider the instance available at `www.cedric.cnam.fr/~lamberta/Library/IQCP_MIQCP/convex_instance.dat`. This instance has 30 binary variables, 1 quadratic and convex inequality constraint and a linear objective function. For this instance we obtain the following results:

- Optimal value of the instance: 0
- Optimal value of the continuous relaxation: -8.18
- Optimal value of the continuous relaxation after reformulation by a complete linearization (i.e. $S_r = \mathbf{0}_n$, $r = 0, \ldots, m$): -34.41
- Optimal value of the continuous relaxation after reformulation by our algorithm: -3.078

The continuous relaxation of this instance is a convex problem. Hence, a first bound b_1 can be computed by solving this problem. Our approach computes a bound b_2 such that $b_2 \geq b_1$ and possibly $b_2 > b_1$ as in this example. Observe that b_1 is the minimum of a linear function subject to one quadratic constraint, while b_2 is the minimum of a quadratic function subject to linear constraints.

4 The case of equality constraints

In this section, we study the case where the initial problem contains some quadratic equality constraints. First, we briefly recall the `MIQCR` method [8] for the purely integer case. This method can handle the following class of problems:

$$(P)\begin{cases} \min & f_0(x) & \\ \text{s.t.} & & \\ & \sum_{i=1}^{n} a_{ri}x_i = b_r & r = 1, \ldots, m \\ & \ell_i \leq x_i \leq u_i & i \in I \\ & x_i \in \mathbb{N} & i \in I \end{cases}$$

By use of a scalar parameter α and of a matrix parameter Φ, we build the following family of quadratic equivalent formulation to (P):

$$(P_{\alpha,\Phi})\begin{cases} \min f_0(x) + \alpha\sum_{r=1}^{m}(\sum_{i=1}^{n} a_{ri}x_i - b_r)^2 + \sum_{i=1}^{n}\sum_{j=i:\Phi_{ij}\neq 0}^{n} \Phi_{ij}(y_{ij} - x_i x_j) & \\ \quad \sum_{i=1}^{n} a_{ri}x_i = b_r & r = 1, \ldots, m \\ (x, Y, z, t) \in P_{xYzt} & \end{cases}$$

We then compute the parameters α^* and Φ^* that give the tightest possible equivalent program to (P), and where the reformulated objective function is convex. These best parameters are deduced from the dual solution of a semidefinite problem that is similar to (SDP).

Below, we highlight the link between our current work and the `MIQCR` method. We show that, in the case of linear equality constraints, our reformulation (QP^*) can be viewed as an extension of the `MIQCR` method. Then, we prove that, unlike for quadratic inequalities, any positive semidefinite matrices can be used to reformulate the quadratic equalities to obtain the tightest equivalent formulation.

4.1 Application of our new method in the presence of quadratic equalities and link to the `MIQCR` method

Without loss of generality, we consider that the two first inequalities in (QP) come from a single equality which was previously rewritten as two inequalities. More formally, the quadratic constraints in (QP) are precisely:

$$\begin{cases} \langle Q_1, xx^T\rangle + c_1^T x \leq b_1 \\ \langle -Q_1, xx^T\rangle - c_1^T x \leq -b_1 \\ \langle Q_r, xx^T\rangle + c_r^T x \leq b_r \qquad\qquad r = 3, \dots m \end{cases}$$

and the two first inequalities come from the equality $\langle Q_1, xx^T\rangle + c_1^T x = b_1$.

The optimally reformulated problem described in Corollary 1 is in this case:

$$(QP^*)\begin{cases} \min \quad f_{0,S_0^*}(x,Y) = \langle S_0^*, xx^T\rangle + c_0^T x + \langle Q_0 - S_0^*, Y\rangle \\ \text{s.t.} \\ \quad \langle Q_1, Y\rangle + c_1^T x \leq b_1 \\ \quad \langle -Q_1, Y\rangle - c_1^T x \leq -b_1 \\ \quad \langle Q_r, Y\rangle + c_r^T x \leq b_r \qquad r = 3, \dots, m \\ \quad (x, Y, z, t) \in P_{xYzt} \end{cases} \tag{23}$$

where

$$S_0^* = Q_0 + (\alpha_1^* - \alpha_2^*)Q_1 + \sum_{r=3}^{m} \alpha_r^* Q_r + \Phi^*$$

and the first two inequalities are equivalent to the equality $\langle Q_1, Y\rangle + c_1^T x = b_1$.

Observation 1 *For any feasible solution (x, Y) to problem (QP^*), we have:*

$$\begin{aligned} f_{0,S_0^*}(x,Y) &= \langle S_0^*, xx^T\rangle + c_0^T x + \langle Q_0 - S_0^*, Y\rangle \\ &= \langle Q_0, xx^T\rangle + c_0^T x + \langle S_0^* - Q_0, xx^T - Y\rangle \\ &= \langle Q_0, xx^T\rangle + c_0^T x + \langle \Phi^*, xx^T - Y\rangle + \sum_{r=3}^{m} \alpha_r^* \langle Q_r, xx^T - Y\rangle \\ &\qquad + (\alpha_1^* - \alpha_2^*)\langle Q_1, xx^T - Y\rangle \\ &= \langle Q_0, xx^T\rangle + c_0^T x + \langle \Phi^*, xx^T - Y\rangle + \sum_{r=3}^{m} \alpha_r^* \langle Q_r, xx^T - Y\rangle \\ &\qquad + (\alpha_1^* - \alpha_2^*)\Big(\langle Q_1, xx^T\rangle + c_1^T x - b_1\Big) \end{aligned}$$

We can notice that the last term of the above calculation is the initial equality constraint multiplied by an unsigned scalar parameter $\nu = \alpha_1^* - \alpha_2^*$. Hence, if equality constraints are considered in the initial formulation and when we build the optimal solution $(S_0^*, \ldots, S_m^*)$ as in Corollary 1, it amounts to explicitely integrating equality constraints into the objective function multiplied by a scalar parameter.

Observation 1 brings us back to the spirit of `MIQCR`. Indeed, in the `MIQCR` method, we integrate in the objective function a set of quadratic functions multiplied by a scalar parameter α. Thus, for the equality case, this new family of convex reformulations can be viewed as an extension of the `MIQCR` method.

4.2 Optimal equivalent reformulations for equality constraints

Here, we start by considering a slightly different reformulation scheme where we explicitly integrate the quadratic inequalities into the objective function. We show that, within this scheme, the equivalent reformulated equalities can be any convex functions.

Without loss of generality, we consider the case of just one equality which corresponds to the first two inequalities. Following the same developments as in Section 3, we obtain the corresponding relaxed problem $(RQP_{\nu,S_0,S_1,S_1',S_3\ldots,S_m})$ where the equality constraint is lifted in the objective function weighted by a scalar ν:

$$\begin{cases} \min f^1(x,Y) = \langle S_0, xx^T\rangle + c_0^T x + \langle Q_0 - S_0, Y\rangle + \nu\Big(\langle Q_1, xx^T\rangle + c_1^T x - b_1\Big) \\ \text{s.t.} \\ \quad \langle S_1, xx^T\rangle + c_1^T x + \langle Q_1 - S_1, Y\rangle \le b_1 & (24) \\ \quad \langle S_1', xx^T\rangle - c_1^T x - \langle Q_1 + S_1', Y\rangle \le -b_1 & (25) \\ \quad \langle S_r, xx^T\rangle + c_r^T x + \langle Q_r - S_r, Y\rangle \le b_r \; r = 3, \ldots, m & (26) \\ \quad (x,Y) \in \bar{P}_{xY} & (27) \end{cases}$$

Finding the best formulation amounts to solving problem (CP'):

$$(CP') \begin{cases} \max\limits_{\substack{\nu \in R,\ S_0+\nu Q_1 \succeq 0 \\ S_1,S_1',S_3\ldots,S_m \succeq 0}} \{v(RQP_{\nu,S_0,S_1,S_1',S_3,\ldots,S_m})\} \end{cases}$$

We prove in Theorem 2 that when quadratic equalities are explicitely integrated in the objective function, in an optimal solution to (CP'), matrices S_1 and S_1' associated to the equality constraint can be any positive semidefinite matrices.

Theorem 2. *Let $(\nu^*, S_0^*, S_1^*, S_1^{'*}, S_3^*, \ldots, S_m^*)$ be an optimal solution to (CP'), then for any semi-definite matrices $\bar{S}_1$, $\bar{S}_1'$, the solution $(\nu^*, S_0^*, \bar{S}_1, \bar{S}_1', S_3^*, \ldots, S_m^*)$ is also an optimal solution to (CP').*

Proof. To prove that, in an optimal solution to (CP'), matrices S_1 and S_1' can be any semi-definite matrices, we prove that solving (CP') is equivalent to solving the following problem (CP^1):

$$(CP^1) \begin{cases} \max\limits_{\substack{\nu\in\mathbb{R} \\ S_0+\nu Q_1\succeq 0 \\ S_3,\ldots,S_m\succeq 0}} \{v(RQP^1_{\nu,S_0,S_3,\ldots,S_m})\} \end{cases}$$

where $(RQP^1_{\nu,S_0,S_3\ldots,S_m})$ is $(RQP_{\nu,S_0,S_1,S_1',S_3\ldots,S_m})$ without Constraints (24) and (25).

$\diamond$ We first prove that $v(CP') \geq v(CP^1)$. This follows from the immediate observation that $v(RQP_{\nu,S_0,S_1,S_1',S_3\ldots,S_m}) \geq v(RQP^1_{\nu,S_0,S_3,\ldots,S_m})$.

$\diamond$ We now prove that $v(CP') \leq v(CP^1)$, or equivalently that $v(DCP') \leq v(CP^1)$, where (DCP') is the following problem:

$$(DCP') \begin{cases} \max\limits_{\substack{\nu\in\mathbb{R} \\ S_0+\nu Q_1\succeq 0 \\ S_1,S_1',S_3,\ldots,S_m\succeq 0 \\ \lambda_1\geq 0\ \lambda_2\geq 0}} \{v(RQP^2_{\nu,S_0,S_1,S_1',S_3\ldots,S_m,\lambda_1,\lambda_2})\} \end{cases}$$

where $(RQP^2_{\nu,S_0,S_1,S_1',S_3\ldots,S_m,\lambda_1,\lambda_2})$ is:

$$\begin{cases} \min f^2(x,Y) = f^1(x,Y) + \lambda_1(\langle S_1, xx^T\rangle + c_1^T x + \langle Q_1 - S_1, Y\rangle - b_1) \\ \qquad +\lambda_2(\langle S_1', xx^T\rangle - c_1^T x - \langle Q_1 + S_1', Y\rangle + b_1) \\ \text{s.t.} \quad (26)(27) \end{cases}$$

(DCP') is obtained from (CP') by dualizing Constraints (24) and (25). By convexity, (DCP') and (CP') have the same optimal values.

Let $(\nu^*, S_0^*, S_1^*, S_1^{'*}, S_3^*, \ldots, S_m^*, \lambda_1^*, \lambda_2^*)$ be an optimal solution to (DCP'). We build the following feasible solution $(\bar{\nu}, \bar{S}_0, \bar{S}_3, \ldots, \bar{S}_m)$ to (CP^1):

- $\bar{S}_0 = S_0^* + \lambda_1^* S_1^* + \lambda_2^* S_1^{'*} + (\lambda_2^* - \lambda_1^*)Q_1$
- $\bar{S}_r = S_r^*$, $r = 3, \ldots, m$
- $\bar{\nu} = \nu^* + \lambda_1^* - \lambda_2^*$

$(\bar{\nu}, \bar{S}_0, \bar{S}_3, \ldots, \bar{S}_m)$ is feasible to (CP^1). Indeed, $\bar{S}_r$, $r = 3, \ldots, m$ are positive semidefinite matrices. Moreover, $\bar{S}_0 + \bar{\nu}Q_1 = S_0^* + \lambda_1^* S_1^* + \lambda_2^* S_1^{'*} + (\lambda_2^* - \lambda_1^*)Q_1 + (\nu^* + \lambda_1^* - \lambda_2^*)Q_1 = S_0^* + \nu^* Q_1 + \lambda_1^* S_1^* + \lambda_2^* S_1^{'*}$ is positive semidefinite since $S_0^* + \nu^* Q_1 \succeq 0$, $S_1^*, S_1^{'*} \succeq 0$ and $\lambda_1^*, \lambda_2^* \geq 0$.

The objective function value of the associated problem $(RQP^1_{\bar{\nu}, \bar{S}_0, \bar{S}_3 \ldots, \bar{S}_m})$ is:

$$\begin{aligned}
f^1(x, Y) &= \langle S_0^* + \lambda_1^* S_1^* + \lambda_2^* S_1^{'*} + (\lambda_2^* - \lambda_1^*)Q_1, xx^T \rangle + c_0^T x \\
&\quad + \langle Q_0 - S_0^* - \lambda_1^* S_1^* - \lambda_2^* S_1^{'*} - (\lambda_2^* - \lambda_1^*)Q_1, Y \rangle \\
&\quad + (\nu^* + \lambda_1^* - \lambda_2^*)(\langle Q_1, xx^T \rangle + c_1^T x - b_1) \\
&= \langle S_0^*, xx^T \rangle + c_0^T x + \langle Q_0 - S_0^*, Y \rangle + \nu^* (\langle Q_1, xx^T \rangle + c_1^T x - b_1) \\
&\quad + \lambda_1^* (\langle S_1^*, xx^T \rangle - \langle S_1^* - Q_1, Y \rangle + c_1^T x - b_1) \\
&\quad + \lambda_2^* (\langle S_1^{'*}, xx^T \rangle - \langle S_1^{'*} + Q_1, Y \rangle - c_1^T x + b_1) \\
&= f^2(x, Y)
\end{aligned}$$

Therefore, $v(RQP^1_{\bar{\nu}, \bar{S}_0, \bar{S}_3 \ldots, \bar{S}_m}) = v(RQP^2_{\nu^*, S_0^*, S_1^*, S_1^{'*}, S_3^*, \ldots, S_m^*, \lambda_1^*, \lambda_2^*})$ as these two problems have the same objective functions and the same set of constraints. □

Let us now state Corollary 2 that, as claimed in the beginning of the section, shows that for any equivalent reformulated equality constraints the value of the best reformulation is reached.

Corollary 2. $v(CP') = v(SDP) = v(CP)$

Proof. We know by Theorem 1 that $v(CP) = v(SDP)$. Moreover, by convexity and by dualizing the first two inequalities of (SDP) which are equivalent to the equality $\langle Q_1, X \rangle + c_1^T x = b_1$, we obtain $v(SDP) = v(CP^1)$. Thus, as $v(CP') = v(CP^1)$ by Theorem 2, we obtain $v(CP') = v(SDP)$. □

5 Extension to the mixed-integer case

In this section, we study the case where some of the variables in the initial problem are continuous. More formally, we aim to solve the following problem:

$$(MQP) \begin{cases} \min f_0(x) & \\ \text{s.t.} & \\ \quad f_r(x) \leq b_r & r = 1, \ldots, m \\ \quad 0 \leq x_i \leq u_i & i \in I \cup J \\ \quad x_i \in \mathbb{N} & i \in I \\ \quad x_i \in \mathbb{R} & i \in J \end{cases}$$

where $\forall\, r \in \{0,\ldots,m\}$, $f_r(x) = \langle Q_r, xx^T\rangle + c_r^T x$, $Q_r \in \mathcal{S}_N$, $c_r \in \mathbb{R}^N$, $b_r \in \mathbb{R}$, $I = \{1,\ldots,n\}$, $J = \{n+1,\ldots,N\}$, and $u_i \in \mathbb{N}\,(i \in I)$, $u_j \in \mathbb{R}\,(j \in J)$.

We denote by $\mathscr{H}$ the set of pairs of indices of variables x involving at least one integer variable:

$$\mathscr{H} = \{(i,j) \in I^2 \cup (I \times J) \cup (J \times I)\}$$

Definition 1. *For any matrix $A \in \mathcal{S}_N$, we define the following decomposition of A into two matrices $M_A \in \mathcal{S}_N$ and $C_A \in \mathcal{S}_N$ such that:*

$$A = M_A + C_A$$

where the terms of M_A, for $(i,j) \in J^2$, are zeros and the terms of C_A, for $(i,j) \in \mathscr{H}$, are zeros. ■

An illustration of Definition 1 is presented in Figure 1. The aim of this definition is to formalize the decomposition of the quadratic function $x^T Ax$ into the sum of the quadratic sub-function with the real variables only $x^T C_A x$ and the other quadratic sub-function $x^T M_A x$. Of course, for any vector x, $x^T Ax = x^T M_A x + x^T C_A x$.

$$A = \begin{array}{|c|c|}\hline A^{I^2} & A^{I\times J} \\ \hline A^{J\times I} & A^{J^2} \\ \hline \end{array} = M_A + C_A = \begin{array}{|c|c|}\hline A^{I^2} & A^{I\times J} \\ \hline A^{J\times I} & 0 \\ \hline \end{array} + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline 0 & A^{J^2} \\ \hline \end{array}$$

Fig. 1. Decomposition of a matrix A into matrices M_A and C_A

Assumption 1 *We make the assumption that for all $r = 0,\ldots,m$, $C_{Q_r} \succeq 0$, i.e. the quadratic sub-functions $x^T C_{Q_r} x$ of real variables are convex functions.* ■

In the following, we first present the extension of Algorithm 1 to the case where $C_{Q_r} = \mathbf{0}_N$ for $r = 1,\ldots,m$, i.e., in the constraints, there are no quadratic sub-functions of real variables. Then, we deduce an algorithm to solve (MQP) in the more general case where the quadratic sub-functions of real variables are convex functions.

5.1 The case where the quadratic sub-functions of real variables of constraints are zero-functions ($C_{Q_0} \succeq 0$, and for $r = 1,\ldots,m$, $C_{Q_r} = \mathbf{0}_N$)

In this section, we suppose that, in the constraints, there are no quadratic terms involving two real variables, i.e. that $\forall r = 1,\ldots,m$, $C_{Q_r} = \mathbf{0}_N$. We now present

our algorithm following the same reasoning steps as for the pure integer case.

As in Section 2, we introduce additional variables y_{ij}, but these variables are defined only for $(i,j) \in \mathscr{H}$ (set of indices of products involving at least one integer variable). Furthermore, we consider semidefinite matrices $S_0, \ldots, S_m \in \mathcal{S}_N$ such that for all $r = 0, \ldots, m$, $S_r = M_{S_r} + C_{S_r}$ with $C_{S_r} = \mathbf{0}_N$.

The following program $(MQP_{S_0,\ldots,S_m})$ is equivalent to (MQP):

$$(MQP_{S_0,\ldots,S_m}) \begin{cases} \min f_{0,S_0}(x,Y) \\ \text{s.t.} \\ \quad f_{r,S_r}(x,Y) \leq b_r & r = 1,\ldots,m \\ \quad (x,Y,z,t) \in P'_{xYzt} \end{cases}$$

where P'_{xYzt} is:

$$P'_{xYzt} \begin{cases} x_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik} & i \in I \\ y_{ij} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k z_{ijk} & (i,j) \in I \times (I \cup J) \\ z_{ijk} \leq u_j t_{ik} & (i,k) \in E,\ j \in I \cup J \\ z_{ijk} \leq x_j & (i,k) \in E,\ j \in I \cup J \\ z_{ijk} \geq x_j - u_j(1 - t_{ik}) & (i,k) \in E,\ j \in I \cup J \\ z_{ijk} \geq 0 & (i,k) \in E,\ j \in I \cup J \\ t_{ik} \in \{0,1\} & (i,k) \in E \\ y_{ii} \geq x_i & i \in I \\ y_{ij} = y_{ji} & (i,j) \in \mathscr{H},\ i < j \\ y_{ij} \geq u_j x_i + u_i x_j - u_i u_j & (i,j) \in \mathscr{H},\ i \leq j \\ y_{ij} \geq 0 & (i,j) \in \mathscr{H},\ i \leq j \end{cases}$$

We now define the semidefinite program (SDP') which is obtained from (SDP) by replacing in Constraints (13)–(16) $(i,j) \in I^2$ by $(i,j) \in \mathscr{H}$. This substitution amounts to not considering these constraints for pairs of real variables.

Theorem 3. *Let (α^*, Φ^*) be an optimal dual solution to (SDP'), where Φ^* is the symmetric matrix built as described in Theorem 1 from the optimal dual variables associated to new Constraints (13)–(16), and α^* is the vector of optimal dual variables associated to Constraints (12). We build S_0^* as follows:*

$$S_0^* = M_{S_0^*} + C_{S_0^*} \text{ where } M_{S_0^*} = M_{Q_0} + \sum_{r=1}^{m} \alpha_r^* M_{Q_r} + M_{\Phi^*} \text{ and } C_{S_0^*} = C_{Q_0}$$

The optimal value of $(MQP^*) = (MQP_{S_0^*, \mathbf{0}_N, ..., \mathbf{0}_N})$ *is equal to the optimal value of* (SDP').

The proof can be deduced from the proof of Theorem 1.

5.2 The case where the quadratic sub-functions of real variables are arbitrary convex functions ($\forall r = 0, \ldots, m$, $C_{Q_r} \succeq 0$)

To handle this case, we propose to first build the problem $(MQP1)$ from (MQP) by replacing constraints

$$c_r^T x + \langle M_{Q_r}, xx^T \rangle + \langle C_{Q_r}, xx^T \rangle \leq b_r \qquad r = 1, ..., m$$

by

$$c_r^T x + \langle M_{Q_r}, xx^T \rangle \leq b_r \qquad r = 1, ..., m$$

Although $(MQP1)$ is a relaxation of (MQP) (as a consequence of Assumption 1), we do not use this fact in the following.

Now, $(MQP1)$ falls in the case of Section 5.1. So, one can apply the reformulation of Section 5.1 to $(MQP1)$. We obtain a program $(MQP1^*)$ equivalent to $(MQP1)$ and whose continuous relaxation is a convex problem.

Finally, in each constraint of $(MQP1^*)$, we add back the quadratic sub-functions of real variables $\langle C_{Q_r}, xx^T \rangle$ and obtain problem (MQP^*) as an equivalent problem to (MQP). More formally, the constraints of (MQP^*) have the following form:

$$c_r^T x + \langle M_{Q_r}, Y \rangle + \langle C_{Q_r}, xx^T \rangle \leq b_r$$

The sequence of transformations is illustrated in Figure 2. The algorithm is presented in Algorithm 2, where Step 1 corresponds to item (i) of Figure 2, Step 2 to item (ii), and Step 3 to item (iii).

5.3 An illustrative example

We consider here the same example as for the pure-integer case (Section 3.3) where we relax the integrality constraint of variable x_4. This example, which we call (MEx), has an optimal value equal to -1884.97 for $x^* = (9, 0, 20, 14.7)$:

$$(MEx) \begin{cases} \min\ f(x) = -4x_1x_2 - 4x_1x_4 + 6x_2x_4 - 3x_3^2 - 2x_3x_4 + 2x_4^2 \\ \text{s.t.} \\ \quad 8x_1^2 + 5x_2^2 + 8x_2x_3 + 4x_2x_4 + 2x_4^2 \leq 1080 \\ \quad 0 \leq x_1 \leq 11 \\ \quad 0 \leq x_2 \leq 14 \\ \quad 0 \leq x_3 \leq 20 \\ \quad 0 \leq x_4 \leq 16 \\ \quad x_i \in \mathbb{N} \qquad i = \{1, 2, 3\} \\ \quad x_4 \in \mathbb{R} \end{cases}$$

(MQP)	(i) $\longrightarrow$	$(MQP1)$
$\Updownarrow (iv)$		$\Updownarrow (ii)$
$(MQP*)$	(iii) $\longleftarrow$	$(MQP1^*)$

Where

(i): Build $(MQP1)$ from (MQP).
(ii): Obtaining $(MQP1^*)$ by the reformulation of Section 5.1.
(iii): Obtaining (MQP^*) by adding to each linearized constraint the quantity $\langle C_{Q_r}, xx^T \rangle$ from the initial constraints.
(iv): The equivalence holds because $(MQP1^*)$ is equivalent to $(MQP1)$ as $Y = xx^T$.

Fig. 2. Illustration of the sequence of transformations

Algorithm 2 Solution algorithm to (MQP) where $\forall r = 0, \ldots, m$, $C_{Q_r} \succeq 0$

step 1: Build $(MQP1)$ from (MQP).

step 2: Reformulation of $(MQP1)$ into $(MQP1^*)$ by the reformulation of Section 5.1 by fixing momentarily C_{Q_r} to $\mathbf{0}_N$, $r = 1, \ldots, m$.

a) Solve the semidefinite program (SDP').
b) Deduce $(S_0^*, \ldots, S_m^*)$ as described in the proof of Theorem 3.

step 3: For each (linearized) constraint of $(MQP1^*)$, add back the initial associated quadratic sub-functions of real variables $\langle C_{Q_r}, xx^T \rangle$, obtaining problem (MQP^*).

step 4: Solve the program (MQP^*) by a MIQP solver.

Following Algorithm 2, we first solve the semidefinite relaxation (SDP'_{MEx}):

$$(SDP'_{MEx}) \begin{cases} \min\ f(X,x) = -4X_{12} - 4X_{14} + 6X_{24} - 3X_{33} - 2X_{34} + 2X_{44} \\ \text{s.t.}\ \ 8X_{11} + 5X_{22} + 8X_{23} + 4X_{24} \leq 1080 \\ \quad X_{ij} \leq u_j x_i \\ \quad X_{ij} \leq u_i x_j \\ \quad -X_{ij} \leq -u_j x_i - u_i x_j + u_i u_j \\ \quad -X_{ij} \leq 0 \\ \quad -X_{ii} \leq -x_i \\ \quad \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \\ \quad x \in \mathbb{R}^4 \quad X \in \mathbf{S}_4 \end{cases}$$

The optimal value of (SDP'_{MEx}) is equal to -2031.995. We obtain the following optimal dual solution:

- $\alpha_1^* = 0$
- $\Phi^* = \begin{pmatrix} 1.89 & 0.51 & -0.21 & 0.49 \\ 0.51 & 2.20 & 0.04 & -0.95 \\ -0.21 & 0.04 & 3.48 & 1.01 \\ 0.49 & -0.95 & 1.01 & 0 \end{pmatrix}$

We build matrix $S_0^* = Q_0 + \alpha_1^* Q_1 + \Phi^*$, and we obtain:

$$S_0^* = \begin{pmatrix} 1.89 & -1.49 & -0.21 & -1.51 \\ -1.49 & 2.20 & 0.04 & 2.05 \\ -0.21 & 0.04 & 0.48 & 0.01 \\ -1.51 & 2.05 & 0.01 & 2 \end{pmatrix} \text{ and thus } Q_0 - S_0^* = \begin{pmatrix} -1.89 & -0.51 & 0.21 & -0.49 \\ -0.51 & -2.20 & -0.04 & 0.95 \\ 0.21 & -0.04 & -3.48 & -1.01 \\ -0.49 & 0.95 & -1.01 & 0 \end{pmatrix}$$

Then, to solve (MEx), we reformulate it into the following quadratic program with a quadratic constraint (MEx^*)

$$(MEx^*) \begin{cases} \min\ f_{0,S_0^*}(x,Y) = -1.89y_{11} - 1.02y_{12} + 0.42y_{13} - 0.98y_{14} - 2.20y_{22} \\ \qquad -0.08y_{23} + 1.9y_{24} - 3.48y_{33} - 2.02y_{34} + 1.89x_1^2 - 2.98x_1x_2 - 0.42x_1x_3 \\ \qquad -3.02x_1x_4 + 2.20x_2^2 + 0.08x_2x_3 + 4.10x_2x_4 + 0.48x_3^2 + 0.02x_3x_4 + 2x_4^2 \\ \text{s.t.}\ \ 8y_{11} + 5y_{22} + 8y_{23} + 4y_{24} + 2x_4^2 \leq 1080 \\ \quad (x,Y,z,t) \in P'_{xYzt} \end{cases}$$

The continuous relaxation value of (MEx^*) is equal to -1910.03, the initial gap is thus of 1.33%.

6 Computational results

In this section, we present experiments on pure-integer (QP) and mixed-integer (MQP) quadratically constrained programs. We randomly generate pure-integer instances $(IQCP_m, m = 0, 1, 5)$ and mixed-integer instances $(MIQCP_m, m = 1)$ where m denotes the number of quadratic inequalities. All instances are available at `www.cedric.cnam.fr/lamberta/Library/iqcpmiqcp.html` [16]. Both for pure- integer $(IQCP_m)$ and mixed integer $(MIQCP_m)$ instances we compare an exact algorithm applied to our best reformulation (Q^P/MQP) and an exact algorithm applied to the complete linearization (LP/LQP) defined by taking $S_r = \mathbf{0}_n$, for $r = 0, ..., m$. More precisely, (LP) and (LQP) consist in linearizing all quadratic terms involving an integer variable and keeping all the

products of two real variables. In the pure integer case, (LP) has a linear objective function and linear constraints. In the mixed case, (LQP) has a quadratic objective function and quadratic constraints, where the quadratic terms of all functions are the initial convex sub-functions of real variables.

Experimental environment:

Our experiments were carried out on a laptop with an Intel core $i7$ processor of 2 GHz and 8 GB of RAM using a Linux operating system.

For solving the semidefinite programs (SDP) and (SDP') we use CSDP [9], where programs are solved with a precision of 10^{-6}.

For solving convex reformulated programs (QP^*), (MQP^*), (LP), and (LQP), we use the solver Cplex version 12.5.0 [14], where programs are solved with a precision of 10^{-8} and a time limit of 3 hours. For our experiments, we use the multithreading mode of Cplex with up to 4 threads.

Legends of Tables 1, 2, 3 and 4:

- *Name*: `c_N_n_i`, where `c` is the class of the instance, `N` is the number of variables, `n` is the number of integer variables and `i` the index of the instance.
- *Opt*: The optimal solution value of the instance or the best known solution which is obtained within the time limit of 3 hours.
- *Cont*: The optimal value of the continuous relaxation of the reformulated problem.
- *Gap*: $\left|\frac{Opt - Cont}{Opt}\right| * 100$.
- *Ref CPU* (Reformulation CPU) : CPU time in seconds for solving (SDP) with CSDP [9].
- *Cplex CPU* : CPU time in seconds for solving (QP^*), (MQP^*), (LP), or (LQP) with Cplex [14].
- *Tot CPU* (Total CPU) : *Ref CPU* + *Cplex CPU*, for reformulations (QP^*) and (MQP^*), and *Cplex CPU* for reformulations (LP) and (LQP). The Total CPU time is limited to 3 hours. If the optimum is not found within this time, we present the final gap $(g\%)$, $g = \left|\frac{Opt - b}{Opt}\right| * 100$ where b is the best bound obtained within the time limit.
- *Nodes*: Number of nodes visited by the branch-and-bound algorithm.

6.1 Computational experiments on pure-integer instances

Instances $(IQCP_m)$ with quadratic inequality constraints ($m = 1$ and $m = 5$)

Our experiments were carried out on inequality constrained pure-integer instances of class $(IQCP_m)$. Each instance consists of minimizing a quadratic

function subject to m quadratic inequality constraints.

$$(IQCP_m)\begin{cases} \min \quad \langle Q_0, xx^T \rangle & \\ \text{s.t.} & \\ \quad x^T Q_r x \leq b_r & \text{r} = 1, \ldots, \text{m} \\ \quad \ell_i \leq x_i \leq u_i & i \in I \\ \quad x_i \in \mathbb{N} & i \in I \end{cases}$$

We generate instances where the coefficients of $(IQCP_m)$ are randomly generated as follows:

- $\ell_i = 0$ and $u_i = 20$, for all $i \in I = \{1, \ldots, n\}$.
- The coefficients of Q_0 are integers uniformly distributed in the interval $[-5, 5]$ with a density of 75%. More precisely, $\forall (i, j) \in I^2 : i \leq j$, we generate $q_{0ij} \in [-5, 5]$, then $q_{0ji} = q_{0ij}$. We observe that the number of negative eigenvalues of the generated matrices is always close to 50%.
- The coefficients of Q_r are integers uniformly distributed in the interval $[0, 10]$ with a density of 25%. More precisely, $\forall (i, j) \in I^2 : i \leq j$, we generate $q_{rij} \in [0, 10]$, then $q_{rji} = q_{rij}$.
- $b_r = \lfloor 0.1 * (\sum_{i=1}^{n} \sum_{j=1}^{n} q_{rij} u_i u_j) \rfloor$.

In these experiments, we fixed $m = 1$ and $m = 5$. For class $(IQCP_1)$, we generate instances with $n = 10, 20, 30$, or 40, and for class $(IQCP_5)$ we generate instances with $n = 10, 20$, or 30. For each n we generate 10 instances obtaining a total of 70 instances.

In table 1, we compare the solution of instances of class $(IQCP_1)$ by reformulations (LP) and (QP^*). We can observe that the initial gap obtained with the reformulation (QP^*) is always much smaller than the gap obtained by the reformulation (LP). More precisely, this gap is divided by a factor of 44 in average over all the instances. As a consequence, the number of nodes visited during the branch-and-bound algorithm is significantly smaller with reformulation (QP^*) in comparison with reformulation (LP). Finally, reformulation (QP^*) is faster than reformulation (LP) for instances with 20 variables or more. Moreover, reformulation (LP) is unable to solve instances of sizes 30 or 40.

In Table 2, we present results of class $(IQCP_5)$. The initial gap is much smaller with (QP^*) than with (LP) since it is divided by a factor of about 13.5. However, this improvement is not as significant as in the case of instances $(IQCP_1)$.We can also notice that the average gap increases with sizes of problems for reformulation (LP), while it remains quite stable for reformulation (QP^*). Regarding the total CPU time, we find that both approaches are comparable

		(QP^*)						(LP)			
	Opt	*Cont*	*Gap*	*Ref CPU*	*Cplex CPU*	*Tot CPU*	*Nodes*	*Cont*	*Gap*	*Tot CPU*	*Nodes*
$IQCP_1$_10_10_1	-6480	-6868.17	5.99	1	1	2	97	-12710.00	96.14	1	122
$IQCP_1$_10_10_2	-8094	-8343.27	3.08	1	1	2	178	-17047.18	110.62	1	489
$IQCP_1$_10_10_3	-9132	-10014.47	9.66	1	5	6	457	-15056.25	64.87	0	248
$IQCP_1$_10_10_4	-12602	-12710.46	0.86	1	0	1	8	-12880.74	2.21	0	0
$IQCP_1$_10_10_5	-16293	-16666.41	2.29	1	1	2	36	-17828.17	9.42	0	0
$IQCP_1$_10_10_6	-10400	-10399.98	0.00	1	1	2	0	-15672.22	50.69	0	18
$IQCP_1$_10_10_7	-19925	-20475.93	2.77	1	1	2	12	-20476.00	2.77	0	0
$IQCP_1$_10_10_8	-17280	-17486.01	1.19	2	0	2	58	-19357.79	12.02	0	0
$IQCP_1$_10_10_9	-8605	-9060.32	5.29	1	1	2	145	-15954.00	85.40	1	177
$IQCP_1$_10_10_10	-18676	-19589.90	4.89	3	2	5	353	-26243.61	40.52	1	641
Average			**3.6**	**1**	**2**	**2.6**	**134**		**47.5**	**0.4**	**169**
$IQCP_1$_20_20_1	-42784	-42930.54	0.34	70	5	75	13	-60173.33	40.64	7	471
$IQCP_1$_20_20_2	-38755	-40134.44	3.56	47	17	64	225	-71456.00	84.38	29	2631
$IQCP_1$_20_20_3	-24332	-24957.35	2.57	52	25	77	298	-59417.14	144.19	250	15094
$IQCP_1$_20_20_4	-42047	-43438.84	3.31	67	46	113	779	-84592.22	101.18	352	18133
$IQCP_1$_20_20_5	-41765	-42768.76	2.40	43	20	63	476	-72682.86	74.03	24	1479
$IQCP_1$_20_20_6	-43168	-43278.08	0.25	69	10	79	131	-70658.57	63.68	30	1182
$IQCP_1$_20_20_7	-40780	-41592.13	1.99	69	41	110	624	-81170.00	99.04	245	9594
$IQCP_1$_20_20_8	-53520	-53742.30	0.42	66	6	72	64	-84250.00	57.42	23	908
$IQCP_1$_20_20_9	-33920	-34161.85	0.71	70	18	88	197	-68750.00	102.68	86	5576
$IQCP_1$_20_20_10	-35412	-37158.67	4.93	71	71	142	1942	-77732.00	119.51	498	19799
Average			**2.0**	**62.2**	**26**	**88.3**	**475**		**88.7**	**154.4**	**7487**
$IQCP_1$_30_30_1	-74564	-75853.65	1.73	328	66	394	258	-155497.50	108.54	(9.6%)	127268
$IQCP_1$_30_30_2	-83240	-83403.34	0.20	468	30	498	29	-181836.00	118.45	(29.8%)	187571
$IQCP_1$_30_30_3	-74176	-74250.59	0.10	177	11	188	9	-171918.00	131.77	(31.8%)	168305
$IQCP_1$_30_30_4	-64252	-66063.23	2.82	479	130	609	700	-143422.00	123.22	(20.5%)	171016
$IQCP_1$_30_30_5	-72564	-74360.49	2.48	359	262	621	1321	-155603.33	114.44	(27.7%)	170972
$IQCP_1$_30_30_6	-68240	-68860.59	0.91	348	148	496	701	-159313.33	133.46	(26.8%)	198873
$IQCP_1$_30_30_7	-80800	-81319.99	0.64	480	28	508	27	-158102.00	95.67	(23.6%)	135321
$IQCP_1$_30_30_8	-65840	-68262.45	3.68	358	761	1119	4034	-160310.00	143.48	(45%)	134769
$IQCP_1$_30_30_9	-58937	-59589.24	1.11	475	65	540	216	-147703.33	150.61	(36.9%)	208016
$IQCP_1$_30_30_10	-69456	-70159.24	1.01	326	54	380	99	-153665.00	121.24	(19.3%)	167622
Average			**1.5**	**379.8**	**155**	**535.3**	**739**		**124.1**	**-**	**166973**
$IQCP_1$_40_40_1	-101328	-103471.51	2.12	1685	3601	5286	4078	-272558.57	168.99	(103.5%)	73878
$IQCP_1$_40_40_2	-134344	-136626.67	1.70	1747	567	2314	1042	-302984.00	125.53	(77.9%)	81929
$IQCP_1$_40_40_3	-118324	-118779.41	0.38	2129	142	2271	75	-316666.25	167.63	(105.8%)	114192
$IQCP_1$_40_40_4	-89641	-95139.22	6.13	2172	(2.7%)	-	29756	-271700.00	203.10	(126.8%)	80102
$IQCP_1$_40_40_5	-111077	-114710.96	3.27	2154	1994	4148	6475	-274985.71	147.56	(84.6%)	69510
$IQCP_1$_40_40_6	-133516	-133814.01	0.22	2189	226	2415	98	-300148.75	124.80	(62.9%)	76622
$IQCP_1$_40_40_7	-105040	-107134.61	1.99	1899	307	2206	542	-267817.14	154.97	(88.6%)	72252
$IQCP_1$_40_40_8	-118782	-119816.45	0.87	2231	619	2850	559	-293542.00	147.13	(81.5%)	73367
$IQCP_1$_40_40_9	-84142	-88195.02	4.82	1541	5203	6744	22650	-279803.33	232.54	(149.4%)	84920
$IQCP_1$_40_40_10	-92809	-96375.14	3.84	2194	2863	5057	7668	-269400.00	190.27	(117.1%)	88860
Average			**2.5**	**1994.1**	**1705**	**3699**	**4799**		**166.3**	**-**	**81563**

Pure integer variables – 1 quadratic inequality constraint - time limit 3 hours

Table 1. Results for the exact solution of instances ($IQCP_1$) by (QP^*) and (LP).

		(QP^*)						(LP)			
	Opt	*Cont*	*Gap*	*Ref CPU*	*Cplex CPU*	*Tot CPU*	*Nodes*	*Cont*	*Gap*	*Tot CPU*	*Nodes*
$IQCP_5$_10_10_5_1	-6880	-7674.31	11.55	3	1	4	108	-11761.69	70.95	1	41
$IQCP_5$_10_10_5_2	-10047	-10743.60	6.93	10	1	11	573	-17503.32	74.21	1	262
$IQCP_5$_10_10_5_3	-7966	-8938.46	12.21	10	1	11	65	-14341.01	80.03	0	327
$IQCP_5$_10_10_5_4	-9799	-10782.58	10.04	7	2	9	608	-19396.44	97.94	1	782
$IQCP_5$_10_10_5_5	-7260	-9469.95	30.44	3	3	6	2007	-17257.59	137.71	1	916
$IQCP_5$_10_10_5_6	-5375	-6419.93	19.44	2	5	7	2903	-14527.34	170.28	2	915
$IQCP_5$_10_10_5_7	-9314	-10063.65	8.05	3	1	4	310	-13127.70	40.95	0	86
$IQCP_5$_10_10_5_8	-7872	-8232.16	4.58	3	1	4	93	-13403.50	70.27	1	99
$IQCP_5$_10_10_5_9	-11716	-12337.24	5.30	2	2	4	256	-17656.32	50.70	1	57
$IQCP_5$_10_10_5_10	-6276	-7738.93	23.31	3	2	5	702	-13818.04	120.17	1	862
Average			**13.2**	**4.6**	**1.9**	**6.5**	**763**		**91.3**	**0.9**	**435**
$IQCP_5$_20_20_5_1	-27747	-29493.40	6.29	103	37	140	2300	-65618.21	136.49	253	15055
$IQCP_5$_20_20_5_2	-36245	-37826.09	4.36	143	24	167	1619	-72305.92	99.49	191	8895
$IQCP_5$_20_20_5_3	-34153	-36884.65	8.00	140	111	251	7701	-74355.19	117.71	268	13950
$IQCP_5$_20_20_5_4	-27466	-29623.50	7.86	141	42	183	3714	-59500.00	116.63	61	5909
$IQCP_5$_20_20_5_5	-24813	-26250.41	5.79	104	37	141	3317	-55442.56	123.44	70	4733
$IQCP_5$_20_20_5_6	-32648	-33236.63	1.80	118	9	127	491	-61143.92	87.28	42	2567
$IQCP_5$_20_20_5_7	-29218	-31178.65	6.71	140	30	170	1607	-61536.10	110.61	148	8830
$IQCP_5$_20_20_5_8	-22016	-24958.28	13.36	140	112	252	10345	-57627.19	161.75	558	40979
$IQCP_5$_20_20_5_9	-33856	-35001.03	3.38	140	16	156	544	-69851.49	106.32	80	4559
$IQCP_5$_20_20_5_10	-32053	-37163.45	15.94	109	554	663	30601	-77604.15	142.11	779	44380
Average			**7.4**	**127.8**	**97.2**	**225**	**6224**		**120.2**	**245**	**14986**
$IQCP_5$_30_30_5_1	-54272	-60373.66	11.24	702	(6.5%)	-	90556	-159296.11	193.51	(70.2%)	140852
$IQCP_5$_30_30_5_2	-53978	-58234.79	7.89	843	5359	6202	83912	-150880.24	179.52	(53.5%)	161325
$IQCP_5$_30_30_5_3	-58168	-61680.35	6.04	840	4167	5007	52184	-148322.72	154.99	(46.5%)	177551
$IQCP_5$_30_30_5_4	-58655	-62121.66	5.91	842	894	1736	7971	-149236.15	154.43	(48.1%)	74678
$IQCP_5$_30_30_5_5	-64708	-69544.03	7.47	692	4713	5405	53807	-166755.83	157.71	(79.6%)	17601
$IQCP_5$_30_30_5_6	-62032	-64632.77	4.19	802	485	1287	7681	-141292.72	127.77	(15.3%)	154769
$IQCP_5$_30_30_5_7	-56226	-60112.81	6.91	872	3059	3931	35313	-142991.58	154.32	(43%)	161049
$IQCP_5$_30_30_5_8	-72261	-76070.89	5.27	782	1136	1918	18873	-166487.75	130.40	(32.8%)	163079
$IQCP_5$_30_30_5_9	-46273	-51609.73	11.53	873	(3%)	-	132604	-140216.73	203.02	(53.2%)	151056
$IQCP_5$_30_30_5_10	-52629	-53716.97	2.07	721	150	871	2128	-126301.11	139.98	(17.1%)	154617
Average			**6.9**	**796.9**	**2495.4**	**3294.6**	**48503**		**159.6**	**-**	**135658**

Pure integer variables – 5 quadratic inequality constraints - time limit 3 hours

Table 2. Results for the solution of instances of class ($IQCP_5$) by (QP^*) and (LP)

for instances with 10 or 20 variables. However, the solution of instances with 30 variables is much faster with reformulation (QP^*) than with reformulation (LP). Thus (QP^*) solves 8 instances of 30 variables, out of 10, while (LP) solves none of them. We do not present results for $n = 40$ because for this size the reformulation phase becomes too difficult with the SDP solvers available to us. Indeed, standards SDP solvers were not able to solve (SDP) for all the considered instances.

Influence of the percentage of negative eigenvalues on our algorithm on instances of class $(IQCP_0)$

We also experiment our algorithm on instances of class $(IQCP_0)$ where we vary the percentage of negative eigenvalues in order to evaluate the impact of this parameter on our algorithm. We generate these instances as in [10]. More precisely, we generate random objective functions as follows: given a percentage $p \in [0, 100]$, we choose n numbers μ_i, where the first $\lfloor pn/100 \rfloor$ are chosen uniformly at random from $[-1, 0]$ and the remaining ones are chosen uniformly at random from $[0, 1]$. Next, we generate n vectors of dimension n each, now choosing all entries uniformly at random from $[-1, 1]$, and orthonormalize them to obtain vectors v_i. The coefficient matrix Q_0 is then calculated as $Q_0 = \sum_{i=1}^{n} \mu_i v_i v_i^T$. Therefore, the parameter p makes it possible to control whether the matrix Q_0 is positive semidefinite ($p = 0$), negative semidefinite ($p = 100$) or indefinite. We randomly generate vector c_0 in $[-1, 1]$, $\ell_i = -10$ and $u_i = 10$ for all $i = 1, \ldots, n$. Then, we replace x_i by $x_i + 10$ in order to obtain $0 \leq x_i \leq 20$. The number of variables n was set to 10, 30, or 50. For each percentage of negative eigenvalues p out of $\{10, 30, 50, 70, 90\}$ and for each n we created 10 instances obtaining a total of 150 instances.

In Table 3 we present results for class $(IQCP_0)$. For this class of problems, we can observe that 147 instances over the 150 considered were solved by reformulation (QP^*) within the time limit of 3 hours. We can first notice that the percentage of negative eigenvalues does not influence the average reformulation time of our algorithm, while the average Cplex time is impacted. We observe that hardest instances are those with 30% and 50% of negative eigenvalues. Indeed, this time is multiplied by about 12 for these instances in comparison to instances with 90% of negative eigenvalues and by about 2 in comparison to instances with 10% and 70% of negative eigenvalues. For these instances, we do not present results of the reformulation (LP), i.e. complete linearisation. Indeed, this method solves only instances of size $n = 10$ within the time limit. Moreover, the average initial gap over all these instances of reformulation (LP) is about 252.5%, while that of reformulation (QP^*) is about 9.5%.

		(QP^*)				
n	p (% negEv)	*Gap*	*Ref CPU*	*Cplex CPU*	*Tot CPU*	*Nodes*
10	10	0.03	0.10	1.00	1.10	34.20
30	10	5.25	113.90	24.70	138.60	7459.80
50	10	8.25	2292.10	1701.30	3993.40	1450943.70
Average		**4.51**	**802.03**	**575.67**	**1377.70**	**486145.90**
10	30	3.19	0.30	0.80	1.10	91.20
30	30	11.66	111.80	58.10	169.90	42979.60
50	30	14.83	2245.70	3643.29 (7)	5877.43 (7)	2360653.86
Average		**9.89**	**785.93**	**1234.06**	**2016.14**	**801241.55**
10	50	14.18	0.20	1.00	1.20	105.60
30	50	10.44	113.30	35.40	148.70	22212.40
50	50	14.76	2271.70	3532.50	5804.20	2434841.60
Average		**13.13**	**795.07**	**1189.63**	**1984.70**	**819053.20**
10	70	6.01	0.30	0.90	1.20	107.70
30	70	9.61	116.00	20.00	136.00	9125.30
50	70	14.21	2209.20	1605.60	3814.80	779952.50
Average		**9.94**	**775.17**	**542.17**	**1317.33**	**263061.83**
10	90	6.09	0.10	1.10	1.20	76.20
30	90	12.78	110.60	13.40	124.00	7154.50
50	90	10.33	2224.30	299.80	2524.10	123295.60
Average		**9.73**	**778.33**	**104.77**	**883.10**	**43508.77**

(i) means that i instances over 10 were solved within the time limit of 3 hours
Pure integer variables – unconstrained - time limit 3 hours

Table 3. Results for the solution of instances of class $(IQCP_0)$ by (QP^*)

6.2 Computational experiments on mixed-integer instances

Our experiments were carried out on inequality constrained mixed-integer instances of class $(MIQCP_m)$. Each instance consists of minimizing a quadratic function subject to m quadratic inequality constraints.

$$(MIQCP_m)\begin{cases} \min \quad \langle Q_0, xx^T\rangle & \\ \text{s.t.} & \\ \quad x^TQ_rx \le b_r & \text{r=1, \ldots, m} \\ \quad \ell_i \le x_i \le u_i & i \in I \cup J \\ \quad x_i \in \mathbb{N} & i \in I \\ \quad x_i \in \mathbb{R} & i \in J \end{cases}$$

For this class, we fixed $m = 1$ (one quadratic inequality), and we generate instances with $13, 27, 40$, and 53 variables, with about $\frac{1}{4}$ of real variables and $\frac{3}{4}$ of integer variables. For $r = 0, 1$, matrices Q_r are equal to $M_{Q_r} + C_{Q_r}$, where M_{Q_r} and C_{Q_r} are defined as in Section 5. The coefficients of $(MIQCP_1)$ are randomly generated as follows:

- $\ell_i = 0$ and $u_i = 10$, for all $i \in I \cup J$.
- The coefficients of M_{Q_0} are integers uniformly distributed in the interval $[-20, 20]$ with a density of 75%. To generate coefficients of C_{Q_0} so that C_{Q_0} is positive semi-definite, we generate a matrix C_A, where the elements of the non-zero block are integers uniformly distributed in the interval $[0, 5]$ with a density of 100%, and $C_{Q_0} = C_AC_A^T$.

- The coefficients of M_{Q_1} are integers uniformly distributed in the interval $[0, 10]$ with a density of 25%. To generate coefficients of C_{Q_1} such that C_{Q_1} is positive semi-definite, we generate a matrix C_A, where the elements of the non-zero block are integers uniformly distributed in the interval $[0, 3]$ with a density of 100%, and $C_{Q_1} = C_A C_A^T$.
- $b_1 = \lfloor 0.1 * (\sum_{i=1}^{n} \sum_{j=1}^{n} q_{1ij} u_i u_j) \rfloor$.

For each $n = 13, 27, 40$, or 53 we generate 10 instances obtaining a total of 40 instances for class $(MIQCP_1)$.

In Table 4, we compare the solution of instances of class $(MIQCP_1)$ by reformulations (LQP) and (MQP^*). These results reveal a similar trend as for class $(IQCP_1)$. More precisely, the initial gap obtained with the reformulation (MQP^*) is much tighter than the gap obtained by the reformulation (LQP) (improved on average by a factor 10), and the number of nodes is smaller with reformulation (MQP^*) in comparison with reformulation (LQP). For these instances, reformulation (LQP) is faster than reformulation (MQP^*) for the smallest instances, with $N = 13, 27$ or 40 when we consider the reformulation time plus the solution time. However, the total solution time of (MQP^*) is highly penalized by the CPU time for solving (SDP'). For larger instances, with $N = 53$, (MQP^*) is faster than (LQP), since it solves all the considered instances in less than 3 hours of CPU time, while reformulation (LQP) is able to only solve 5 instances out of 10.

7 Conclusion

In this paper, we considered the general problem of minimizing a quadratic function subject to equality and/or inequality quadratic constraints. Variables can be integer or continuous but the sub-functions of pure continuous variables must be convex. The general idea of our approach is to reformulate this problem as an equivalent quadratic program whose continuous relaxation is convex. This makes it easier to globally solve the problem by branch-and-bound. We show that the reformulation which provides the best continuous relaxation is obtained by solving a semidefinite relaxation of the original problem.

We report computational results on 220 instances. These results show that in the pure integer case the method allows us to solve almost all the considered instances with up to 40 variables and one or five inequality quadratic constraints in less than three hours of computation time. Regarding the mixed-integer case, the method can handle instances with 40 integer variables, 13 continuous variables, and one inequality constraint in less than three hours of computation time.

A perspective of this work is to extend our approach to any mixed-integer quadratic program.

		(MQP^*)						(LQP)			
	Opt	*Cont*	*Gap*	*Ref CPU*	*Cplex CPU*	*Tot CPU*	*Nodes*	*Cont*	*Gap*	*Tot CPU*	*Nodes*
$MIQCP_1$_13_10_1	-4764.13	-4764.82	0.01	1	1	2	58	-6574.15	37.99	1	364
$MIQCP_1$_13_10_2	-13559.99	-13618.17	0.43	1	1	2	47	-13824.85	1.95	0	51
$MIQCP_1$_13_10_3	-18699.99	-18699.95	0.00	1	0	1	0	-18700.00	0.00	1	51
$MIQCP_1$_13_10_4	-13781.29	-14500.63	5.22	1	1	2	256	-14938.91	8.40	1	108
$MIQCP_1$_13_10_5	-6288.95	-6399.57	1.76	1	1	2	223	-8558.21	36.08	1	286
$MIQCP_1$_13_10_6	-4909.99	-4928.72	0.38	1	0	1	11	-6189.20	26.05	0	39
$MIQCP_1$_13_10_7	-6682.81	-7129.04	6.68	1	1	2	29	-9230.19	38.12	0	180
$MIQCP_1$_13_10_8	-8337.85	-9874.30	18.43	1	1	2	139	-12221.75	46.58	0	510
$MIQCP_1$_13_10_9	-5699.99	-5699.95	0.00	1	0	1	0	-8596.87	50.82	0	39
$MIQCP_1$_13_10_10	-14867.07	-15087.38	1.48	1	1	2	41	-16644.56	11.96	0	131
Average			**3.4**	**1.1**	**0.7**	**1.8**	**80**		**25.8**	**0.4**	**176**
$MIQCP_1$_27_20_1	-18911.98	-18919.68	0.04	59	3	62	188	-21448.27	13.41	3	230
$MIQCP_1$_27_20_2	-20733.04	-20733.87	0.00	66	2	68	5	-23262.06	12.20	3	53
$MIQCP_1$_27_20_3	-15766.99	-15818.26	0.33	71	6	77	1018	-23353.64	48.12	9	1628
$MIQCP_1$_27_20_4	-26393.34	-27129.78	2.79	63	4	67	150	-31457.11	19.19	8	958
$MIQCP_1$_27_20_5	-19299.99	-19299.88	0.00	61	2	63	30	-19573.13	1.42	2	0
$MIQCP_1$_27_20_6	-20664.00	-20664.25	0.00	72	3	75	54	-24569.77	18.90	4	459
$MIQCP_1$_27_20_7	-31399.46	-31647.82	0.79	50	3	53	34	-32810.55	4.49	4	143
$MIQCP_1$_27_20_8	-13499.96	-13677.20	1.31	65	4	69	275	-18603.78	37.81	15	2633
$MIQCP_1$_27_20_9	-21905.47	-22361.32	2.08	66	2	68	22	-27385.42	25.02	7	989
$MIQCP_1$_27_20_10	-14687.99	-14701.87	0.09	79	5	84	666	-21432.47	45.92	11	2016
Average			**0.7**	**65.2**	**3.4**	**68.6**	**244**		**22.7**	**6.6**	**911**
$MIQCP_1$_40_30_1	-33303.89	-37559.88	12.78	596	64	660	6015	-46755.00	40.39	1052	54209
$MIQCP_1$_40_30_2	-34323.98	-34324.49	0.00	849	8	857	10	-45130.00	31.48	70	2181
$MIQCP_1$_40_30_3	-45993.24	-46807.64	1.77	887	19	906	929	-58554.29	27.31	195	12228
$MIQCP_1$_40_30_4	-41227.94	-42895.96	4.05	774	10	784	242	-52083.78	26.33	300	8848
$MIQCP_1$_40_30_5	-28947.86	-30127.91	4.08	790	10	800	186	-39883.60	37.78	219	8930
$MIQCP_1$_40_30_6	-19780.97	-19784.60	0.02	825	20	845	1887	-27558.67	39.32	270	17435
$MIQCP_1$_40_30_7	-36047.02	-36046.76	0.00	754	7	761	2	-45425.35	26.02	80	2405
$MIQCP_1$_40_30_8	-28687.92	-28784.52	0.34	867	6	873	14	-39845.43	38.89	175	9966
$MIQCP_1$_40_30_9	-23702.93	-24703.67	4.22	729	13	742	810	-32877.05	38.70	131	7086
$MIQCP_1$_40_30_10	-29195.98	-29196.39	0.00	747	10	757	129	-38001.92	30.16	90	3671
Average			**2.7**	**781.7**	**16.7**	**798.4**	**1022**		**33.6**	**258.2**	**12696**
$MIQCP_1$_53_40_1	-40830.98	-42804.81	4.83	3632	336	3968	12233	-55848.83	36.78	(7.4%)	104560
$MIQCP_1$_53_40_2	-36483.82	-39341.73	7.83	4623	107	4730	5619	-58532.53	60.43	(14.1%)	335258
$MIQCP_1$_53_40_3	-45280.96	-45281.28	0.00	4274	93	4367	582	-59848.30	32.17	3548	50664
$MIQCP_1$_53_40_4	-36057.83	-37009.76	2.64	3824	11	3835	506	-53618.14	48.70	8778	230101
$MIQCP_1$_53_40_5	-36545.36	-36545.94	0.00	3362	26	3388	43	-45177.88	23.62	734	9657
$MIQCP_1$_53_40_6	-57271.95	-58418.01	2.00	4224	84	4308	2169	-74298.57	29.73	8817	114865
$MIQCP_1$_53_40_7	-31112.93	-37194.08	19.55	3223	2742	5965	168641	-59618.12	91.62	(37.3%)	156685
$MIQCP_1$_53_40_8	-36324.93	-38960.41	7.26	3576	335	3911	12831	-52440.28	44.36	(6.6%)	134298
$MIQCP_1$_53_40_9	-40052.92	-45408.84	13.37	3821	411	4232	31694	-66801.44	66.78	(27.4%)	144419
$MIQCP_1$_53_40_10	-48979.76	-49218.53	0.49	4354	107	4461	2486	-58133.68	18.69	1378	55975
Average			**5.8**	**3891.3**	**425.2**	**4316.5**	**23680**		**45.3**	**4651**	**133648**

Mixed-integer variables – 1 quadratic inequality constraint - time limit 3 hours

Table 4. Results for the solution of instances of class ($MIQCP_1$) by (MQP^*) and (LQP)

References

1. W. P. Adams and H. D. Sherali. Reformulation linearization technique for discrete optimization problems. *Panos M. Pardalos, Ding-Zhu Du, and Ronald L. Graham, editors, Handbook of Combinatorial Optimization,Springer New York*, pages 2849–2896, 2013.
2. F. A. Al-Khayyal, C. Larsen, and T. Van Voorhis. A relaxation method for nonconvex quadratically constrained programs. *Journal of Global Optimization*, 6:215–230, 1995.
3. M.F. Anjos and J.B. Lasserre. Handbook of semidefinite, conic and polynomial optimization: Theory, algorithms, software and applications. *International Series in Operational Research and Management Science*, 166, 2012.
4. C. Audet, P. Hansen, B. Jaumard, and G. Savard. A branch and cut algorithm for non-convex quadratically constrained quadratic programming. *Mathematical Programming*, 87(1):131–152, 2000.
5. C. Audet, P. Hansen, and G. Savard. Essays and surveys in global optimization. *GERAD 25th Anniversary Series, Springer, New York*, 2005.
6. X. Bao, N.V. Sahinidis, and M. Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129:129–157, 2011.
7. A. Billionnet, S. Elloumi, and A. Lambert. Linear reformulations of integer quadratic programs. In *MCO 2008, september 8-10*, pages 43–51, 2008.
8. A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to the case of general mixed integer program. *Mathematical Programming*, 131(1):381–401, 2012.
9. B. Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.
10. C. Buchheim and A. Wiegele. Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141(1–2):435–452, 2013.
11. S. Burer and A. Letchford. Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Oper. Res. and Mgmt. Sci.*, 17(2):97–106, 2012.
12. C. A. Floudas. Deterministic Global Optimization. *Kluwer Academic Publishing, Dordrecht, The Netherlands*, 2000.
13. M.R. Garey and D.S. Johnson. Computers and Intractability: A guide to the theory of NP-Completness. *W.H. Freeman, San Francisco, CA*, 1979.
14. IBM-ILOG. Ibm ilog cplex 12.5 reference manual. "`http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r2/index.jsp`", 2013.
15. A. Lambert. IQCP/MIQCP: Library of Integer and Mixed-Integer Quadratic Quadratically Constrained Programs. "`http://cedric.cnam.fr/~lambe_a1/smiqp/Library/iqcp_miqcp.html`", 2013.
16. L. Liberti and N. Maculan. Global optimization: From theory to implementation, chapter: Nonconvex optimization and its applications. *Springer, New York*, 2006.
17. J. Linderoth. A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103:251–282, 2005.
18. G.P. McCormick. Computability of global solutions to factorable non-convex programs: Part i - convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.
19. R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming*, 136:155–182, 2012.

20. R. Misener and C. A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50, 2013.
21. U. Raber. A simplicial branch-and-bound method for solving nonconvex all-quadratic programs. *Journal of Global Optimization*, 13:417–432, 1998.
22. A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, 130:359–413, 2011.
23. M. Tawarmalani and N.V. Sahinidis. Convexification and global optimization in continuous and mixed-integer nonlinear programming. *Kluwer Academic Publishing, Dordrecht, The Netherlands*, 2002.
24. M. Tawarmalani and N.V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005.