

Extending the QCR method to general mixed-integer programs

Alain Billionnet¹, Sourour Elloumi², and Amélie Lambert²

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. Let (MQP) be a general mixed integer quadratic program that consists of minimizing a quadratic function subject to linear constraints. In this paper, we present a convex reformulation of (MQP) , i.e. we reformulate (MQP) into an equivalent program, with a convex objective function. Such a reformulation can be solved by a standard solver that uses a branch and bound algorithm.

We prove that our reformulation is the best one within a convex reformulation scheme, from the continuous relaxation point of view. This reformulation, that we call MIQCR (Mixed Integer Quadratic Convex Reformulation), is based on the solution of an SDP relaxation of (MQP) . Computational experiences are carried out with instances of (MQP) including one equality constraint or one inequality constraint. The results show that most of the considered instances with up to 40 variables can be solved in one hour of CPU time by a standard solver.

Key words: General integer programming, mixed-integer programming, quadratic programming, convex reformulation, semi-definite programming, experiments

1 Introduction

Consider the following linearly-constrained mixed-integer quadratic program:

$$(MQP) \begin{cases} \text{Min } h(x) \\ \text{s.t. } Ax = b & (1) \\ Dx \leq e & (2) \\ 0 \leq x_i \leq u_i \quad i \in I & (3) \\ 0 \leq x_i \leq u_i \quad i \in J & (4) \\ x_i \in \mathbb{N} \quad i \in I & (5) \\ x_i \in \mathbb{R} \quad i \in J & (6) \end{cases}$$

where $A \in \mathbf{M}_{m,N}$ (set of $m \times N$ matrices), $b \in \mathbb{R}^m$, $D \in \mathbf{M}_{p,N}$, $e \in \mathbb{R}^p$, $I = \{1, \dots, n\}$ is the subset of indices of integer variables, $J = \{n+1, \dots, N\}$ is the subset of indices of real variables, $u_i \in \mathbb{N}$ ($i \in I$), $u_i \in \mathbb{R}$ ($i \in J$), and

$$h(x) = x^T Q x + c^T x = \sum_{(i,j) \in I^2} q_{ij} x_i x_j + \sum_{(i,j) \in I \times J} 2q_{ij} x_i x_j + \sum_{(i,j) \in J^2} q_{ij} x_i x_j + \sum_{i \in I \cup J} c_i x_i.$$

The quadratic sub-function of real variables, $\sum_{(i,j) \in J^2} q_{ij}x_i x_j$, is assumed to be a convex function, $Q \in \mathbf{S}_N$ (space of symmetric matrices of order N), and $c \in \mathbb{R}^N$. Without loss of generality, we shall suppose the feasible domain of (MQP) non-empty.

When the subset of indices I of integer variables is empty, (MQP) is a continuous quadratic convex problem which solution is polynomial. On the other hand, when the subset of indices J of the real variables is empty, (MQP) is a general integer quadratic problem which solution is \mathcal{NP} -hard [12]. To the best of our knowledge, very few publications consider solution of (MQP) in this last case [19, 20, 25].

Many applications in operations research and industrial engineering involve discrete variables in their formulation. Some of these applications can be formulated as (MQP) . For instance, (MQP) is used in [10] for the unit commitment problem and for the Markowitz mean-variance model, in [11] for the chaotic mapping of complete multipartite graphs, in [7] for the material cutting, and in [17] for the capacity planning.

(MQP) belongs to the class of Mixed Integer Non Linear Programs (MINLP). These problems are \mathcal{NP} -hard [12]. Conventional approaches to solve (MINLP) include heuristic methods and global optimization techniques. Usually, heuristic methods consist of reducing as much as possible the non-convexity difficulty. The second approach consists of applying a Branch and Bound algorithm. The success of this last approach depends on the branching rules applied, that try to improve the bound for every sub-problem [1, 8, 21, 26]. The major drawback of applying one of these two approaches for solving (MQP) is that they are very general and thus less appropriate in the specific case of (MQP) which objective function is quadratic.

Standard solvers [4, 18] can efficiently solve Mixed Integer Quadratic Programs (MIQP), but only in the specific case where $h(x)$ is convex. Thus, to solve (MQP) by use of a standard solver, we choose to reformulate it into another program with a convex objective function. By convex reformulation, we mean to design a program, that is equivalent to (MQP) , and that has a convex objective function. In concrete terms, that consists of perturbing the Q matrix of $h(x)$ in order to obtain a positive semi-definite matrix. Further, we will focus on convex reformulations that lead to tight continuous relaxation bounds.

Binary quadratic programming is a particular case of (MQP) , where J is empty and upper bounds u_i are equal to 1. Our contribution in this work is to extend the ideas of QCR (Quadratic Convex Reformulation) [2, 3] from the binary case to the general mixed-integer case. We will see that, by construction, our approach is also an improvement of QCR even in the binary case. First, it improves QCR in terms of bound obtained by continuous relaxation, and second it allows to solve a larger class of problems, including mixed 0-1 quadratic programming.

The outline of the paper is the following. In Section 2, we recall the QCR approach. Then, in the rest of the paper, we present our approach, that we called MIQCR (Mixed Integer Quadratic Convex Reformulation). In Section 3, we present our extension of QCR to general integer programming. Then, in Section 4, we adapt our extension to general mixed-integer programming. In Section 5, we show how to use the inequality constraints to perturb the objective function of the reformulated problem. Finally, in Section 6, we evaluate MIQCR from the computational point of view. Our experiments are carried out on instances of (MQP) including one equality constraint or one inequality constraint. These two different problems are called here the Equality Integer Quadratic Problem (EIQP), and the Inequality Integer Quadratic Problem (IIQP), respectively. Section 7 is a conclusion.

2 The original QCR (Quadratic Convex Reformulation) method [3]

In previous works, convex reformulations were introduced for binary quadratic programming. Indeed, it is easy to convexify a binary quadratic program because we have $x_i^2 = x_i$. Hammer and Rubin [15] use this property to add $-\lambda_{\min}(Q)\sum_{i=1}^n(x_i^2 - x_i)$ to the objective function. This addition amounts to subtracting from the diagonal terms of matrix Q its smallest eigenvalue $\lambda_{\min}(Q)$, what is sufficient to get a positive semi-definite matrix. In [2], Billionnet and Elloumi improve the smallest eigenvalue approach. Again using the property $x_i^2 = x_i$, they introduce a convexification scheme that modifies each diagonal term of Q by a different coefficient. Then they search for an optimal convexification, in the sense that it produces the best bound obtained by continuous relaxation. They show that the optimal coefficients can be deduced from the dual solution of an SDP relaxation of the initial problem.

Then, Billionnet, Elloumi and Plateau [3] extend this last approach to the equality constrained 0-1 quadratic programs in a method that they called QCR. Let (QP^{01}) be the following equality constrained 0-1 quadratic program:

$$(QP^{01}) \begin{cases} \text{Min } f(x) \\ \text{s.t. } \sum_{j=1}^n a_{rj}x_j = b_r \quad r = \{1, \dots, m\} \\ x \in \{0, 1\}^n \end{cases}$$

where

$$f(x) = x^T Q x + c^T x.$$

The idea of QCR is to perturb not only the diagonal terms of Q by a vector parameter, but also the off-diagonal ones using a matrix parameter. In addition to $x_i^2 - x_i$, the authors add to the objective function quadratic functions

that vanish on the feasible domain. These new functions are obtained by multiplying the linear equality constraints by the variables. Thus, from a constraint $\sum_{j=1}^n a_{rj}x_j = b_r$, they build the functions $\sum_{j=1}^n a_{rj}x_jx_i - b_rx_i$.

In other words, the authors consider the following 0-1 quadratic problem equivalent to (QP^{01}) and depending on two parameters ρ and σ :

$$(QP_{\rho,\sigma}^{01}) \begin{cases} \text{Min } f_{\rho,\sigma}(x) \\ \text{s.t. } \sum_{j=1}^n a_{rj}x_j = b_r \quad r = \{1, \dots, m\} \\ x \in \{0, 1\}^n \end{cases}$$

where

$$f_{\rho,\sigma}(x) = f(x) + \sum_{i=1}^n \sigma_i(x_i^2 - x_i) + \sum_{r=1}^m \left(\sum_{i=1}^n \rho_{ri}x_i \right) \left(\sum_{j=1}^n a_{rj}x_j - b_r \right)$$

It is easy to verify that $f_{\rho,\sigma}(x) = f(x)$ in the domain of x .

Let (SDP^{01}) be the following semi-definite program:

$$(SDP^{01}) \begin{cases} \text{Min } f(X, x) = \sum_{i=1}^n \sum_{j=1}^n q_{ij}X_{ij} + \sum_{i=1}^n c_i x_i \\ \text{s.t. } \sum_{j=1}^n a_{rj}X_{ij} - b_r x_i = 0 \quad i \in I, \quad r = \{1, \dots, m\} \quad (7) \\ X_{ii} = x_i \quad i \in I \quad (8) \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 \quad (9) \\ x \in \mathbb{R}^n \quad X \in \mathbf{S}_n \quad (10) \end{cases}$$

The authors search for values of ρ and σ such that both make $f_{\rho,\sigma}(x)$ convex and maximize the value of the continuous relaxation of $(QP_{\rho,\sigma}^{01})$. They prove that the optimal values ρ_{ri}^* of ρ_{ri} ($i \in I, r = \{1, \dots, m\}$) are given by the optimal values of the dual variables associated with constraints (7), and the optimal values σ_i^* of σ_i ($i \in I$) are given by the optimal values of the dual variables associated with constraints (8).

Practically, **QCR** outperforms the method in [2] for equality constrained programs because it integrates equality constraints in the convexification process.

Note that it can be deduced from [9] (Proposition 5) that it is equivalent, from the continuous relaxation point of view, to perturb the objective function with equality constraints using a matrix parameter ρ or a scalar one α . One can think that a better bound would be obtained by adding the $m \times n$ expressions $\rho_{ri}x_i \left(\sum_{j=1}^n a_{rj}x_j - b_r \right)$ with their own multiplier ρ_{ri} . But, it is easy to see that it is

equivalent to perturb the objective function by adding $\alpha \sum_{r=1}^m (\sum_{j=1}^n a_{rj}x_j - b_r)^2$, i.e.

by considering the function $f_{\alpha,\sigma}(x) = f(x) + \sum_{i=1}^n \sigma_i(x_i^2 - x_i) + \alpha \sum_{r=1}^m (\sum_{j=1}^n a_{rj}x_j - b_r)^2$.

Indeed, if $f_{\rho^*,\sigma^*}(x)$ is the best convex function equivalent to $f(x)$, then $f_{\alpha^*,\sigma^*}(x)$, with $\alpha^* = \max_{r,i} \{\rho_{ri}^* : r = 1, \dots, m, i = 1, \dots, n\}$, is also a convex function that is equal to $f_{\rho^*,\sigma^*}(x)$ on the feasible set of the continuous relaxation of (QP^{01}) .

In the rest of the paper we propose an extension of QCR to general mixed-integer quadratic programming.

3 Dealing with general integer variables: the pure integer case

In the first part of this work, we propose a convex reformulation of a specific case of (MQP) where all the variables are integers, i.e. $J = \emptyset$. Let (QP) be a such program:

$$(QP) \begin{cases} \text{Min } f(x) \\ \text{s.t. } (1)(2)(3)(5) \end{cases}$$

where $f(x)$ is the following non-convex function:

$$f(x) = x^T Qx + c^T$$

and $Q \in \mathbf{S}_n$, $c \in \mathbb{R}^n$, $A \in \mathbf{M}_{m,n}$, $b \in \mathbb{R}^m$, $D \in \mathbf{M}_{p,n}$, $e \in \mathbb{R}^p$, $u \in \mathbb{N}^n$.

As there exists an abundant literature on binary quadratic programming, for example QCR [2, 3] or [6, 13–16], a natural way to solve (QP) is to replace each integer variable by an expression of binary ones. Unfortunately, the straightforward reformulation of an integer program into a binary one increases drastically the size of the reformulated problem. Thus, this approach is not efficient for large problems.

In the integer case, the main difference with binary quadratic programming is that x_i^2 is no longer equal to x_i . However, we can extend the smallest eigenvalue approach by adding to the objective function $-\lambda_{\min}(Q) \sum_{i=1}^n (x_i^2 - v_i)$ where $v_i = \sum_{k=0}^{u_i} k^2 r_{ik}$ and r_{ik} are additional binary variables satisfying $\sum_{k=0}^{u_i} r_{ik} = 1$ and $x_i = \sum_{k=0}^{u_i} k r_{ik}$. The obtained program has a convex quadratic function, $\sum_{i=1}^n (1 + u_i)$ new binary variables, and $3n$ new constraints. This provides us with a first method for solving (QP) that we call NC (Naive Convexification).

In this section, we propose a reformulation scheme of (QP) into an equivalent integer quadratic program $(QP_{\alpha,\beta})$ depending on a scalar parameter α , and on a matrix parameter β . We prove that the continuous relaxation of $(QP_{\alpha,\beta})$ is equivalent to another program $(P_{\alpha,\beta})$ having a reduced size. Then, using $(P_{\alpha,\beta})$, we look for α^* and β^* , the values of α and β that maximize the optimal value of $(P_{\alpha,\beta})$. We show that α^* and β^* can be deduced from the solution of a semi-definite relaxation of (QP) .

3.1 A convex reformulation scheme

In our extension of QCR [2, 3] to the general integer case, we introduce new variables y_{ij} , and new linear constraints to enforce the equality $y_{ij} = x_i x_j$. These new variables will allow each term of matrix Q to be perturbed. In order to enforce the equality $y_{ij} = x_i x_j$, with a reduced number of additional variables, we use the binary decomposition of the x_i variables rather than the unary decomposition of \mathbf{NC} . In a sense, our approach mixes ideas of linearization and convexification.

More precisely, given parameters α and β , we build $(QP_{\alpha,\beta})$, a problem equivalent to (QP) . $(QP_{\alpha,\beta})$ can be viewed as a compact transformation of (QP) into a mixed binary quadratic program. We then prove a projection property of the polyhedron defined in $(QP_{\alpha,\beta})$. This projection property will allow us to prove in Section 3.2 that optimal values of parameters α and β can be deduced from a semi-definite relaxation of (QP) . An interesting fact is that this semi-definite relaxation is only built from the initial x variables.

Let us rewrite (QP) into $(QP_{\alpha,\beta})$. The idea is to add to the initial objective function $f(x)$ the following functions that vanish on the feasible domain of (QP) under the assumption that $y_{ij} = x_i x_j$:

- $\alpha \|Ax - b\|^2$ where $\alpha \in \mathbb{R}$.
- $\langle \beta, xx^T - y \rangle$ where $\beta \in \mathbf{S}_n$ and $\langle V, W \rangle$ stands for $\sum_{i=1}^n \sum_{j=1}^n v_{ij} w_{ij}$.

We obtain the following program $(QP_{\alpha,\beta})$:

$$(QP_{\alpha,\beta}) \begin{cases} \text{Min } f_{\alpha,\beta}(x, y) \\ \text{s.t. } (1)(2) \\ x, y, z, t \in P_{xyz} \end{cases}$$

where $f_{\alpha,\beta}(x, y)$ is the following function:

$$f_{\alpha,\beta}(x, y) = f(x) + \alpha \|Ax - b\|^2 + \langle \beta, xx^T - y \rangle.$$

and P_{xyzt} is the following set:

$$P_{xyzt} \left\{ \begin{array}{l}
 (3) \\
 x_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik} \quad i \in I \quad (11) \\
 z_{ijk} \leq u_j t_{ik} \quad (i, k) \in E, j \in I \quad (12) \\
 z_{ijk} \leq x_j \quad (i, k) \in E, j \in I \quad (13) \\
 z_{ijk} \geq x_j - u_j(1 - t_{ik}) \quad (i, k) \in E, j \in I \quad (14) \\
 z_{ijk} \geq 0 \quad (i, k) \in E, j \in I \quad (15) \\
 y_{ij} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k z_{ijk} \quad (i, j) \in I^2 \quad (16) \\
 t_{ik} \in \{0, 1\} \quad (i, k) \in E \quad (17) \\
 y_{ij} = y_{ji} \quad (i, j) \in I^2 \quad (18) \\
 y_{ij} \geq x_i u_j + x_j u_i - u_i u_j \quad (i, j) \in I^2 \quad (19) \\
 y_{ii} \geq x_i \quad i \in I \quad (20)
 \end{array} \right.$$

with $E = \{(i, k) : i = 1, \dots, n, k = 0, \dots, \lfloor \log(u_i) \rfloor\}$

Let us prove that $(QP_{\alpha, \beta})$ is equivalent to (QP) . First, it is straightforward that if x satisfies constraints (1) and $x_i x_j = y_{ij}$, then $f_{\alpha, \beta}(x, y) = f(x)$. Thus, it is enough to prove that the linear constraints of P_{xyzt} enforce the equality between $x_i x_j$ and y_{ij} .

To build P_{xyzt} we use the unique binary decomposition $x_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik}$.

Then we deduce the equality $x_i x_j = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k t_{ik} x_j = y_{ij}$, that we linearize by introducing new variables z_{ijk} to replace each quadratic term $t_{ik} x_j$. Then we add the set of inequalities (12)-(15) ensuring that the z_{ijk} are equal to the $t_{ik} x_j$. To prove that $z_{ijk} = t_{ik} x_j$, it is sufficient to consider the two different values of t_{ik} in (12)-(15).

Constraints (18), (19) and (20) are redundant, but they are necessary for the projection property of Theorem 1. Constraints (18) follow from the equality $x_i x_j = x_j x_i$. We build constraints (19) by multiplying constraints (3) together: $(x_i - u_i)(x_j - u_j) \geq 0$, as they were introduced by McCormick [22]. Constraints (20) follow from the inequality $x_i^2 \geq x_i$ that is satisfied for any integer x_i .

Our reformulated program $(QP_{\alpha, \beta})$ has $\sum_{i=1}^n (1 + \log(u_i))$ new binary variables, $n(n + \sum_{i=1}^n (1 + \log(u_i)))$ real variables, and $3n \sum_{i=1}^n (1 + \log(u_i)) + 3n^2 + 2n$ new constraints.

In this paper, we are interested in the reformulations of (QP) into $(QP_{\alpha, \beta})$ if $f_{\alpha, \beta}(x, y)$ is convex. This is always possible. For example, if we denote by $\lambda_{\min}(Q)$ the smallest eigenvalue of Q , by taking $\beta_{ii} = -\lambda_{\min}(Q)$, $\beta_{ij} = 0$ for all

$i \neq j$, and $\bar{\alpha}$ any non-negative real, we build the convex function $f_{\bar{\alpha},\bar{\beta}}(x,y)$.

We now search for the values of α and β that both make $f_{\alpha,\beta}(x,y)$ convex and maximize the continuous relaxation value of $(QP_{\alpha,\beta})$, that we denote by $(\overline{QP}_{\alpha,\beta})$:

$$(\overline{QP}_{\alpha,\beta}) \begin{cases} \text{Min } f_{\alpha,\beta}(x,y) \\ \text{s.t. } (1)(2) \\ x,y,z,t \in \overline{P}_{xyzt} \end{cases}$$

where \overline{P}_{xyzt} is the polyhedron obtained by relaxing the integrality constraints of P_{xyzt} , i.e. replacing constraints (17) by:

$$0 \leq t_{ik} \leq 1 \quad (17)'$$

In the following, we prove that $(\overline{QP}_{\alpha,\beta})$ is equivalent to another program with less variables and constraints.

Theorem 1. *The projection of \overline{P}_{xyzt} on variables x and y is the following polyhedron:*

$$P_{xy} \begin{cases} (18)(19)(20) \\ x,y : \begin{cases} y_{ij} \geq 0 & (i,j) \in I^2 \quad (21) \\ y_{ij} \leq u_j x_i & (i,j) \in I^2 \quad (22) \\ y_{ij} \leq u_i x_j & (i,j) \in I^2 \quad (23) \end{cases} \end{cases}$$

Proof.

1. We first prove that for any (x,y) in P_{xy} , there exists z,t such that (x,y,z,t) belongs to \overline{P}_{xyzt} .
Let $(x,y) \in P_{xy}$, and γ_{ij} be any non-negative real such that $y_{ij} = \gamma_{ij} x_i x_j$.
Let $\bar{u}_i = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k$. Take $t_{ik} = x_i / \bar{u}_i$ and $z_{ijk} = \gamma_{ij} t_{ik} x_j$. Obviously, constraints (3), (11), (15), (16), (17'), (18), (19) and (20) are satisfied. We have now to prove that constraints (12), (13) and (14) are satisfied:
 - Constraints (12): if $x_i = 0$, then $t_{ik} = 0$, and $z_{ijk} = 0$. If $x_i > 0$, it is sufficient to show that $\gamma_{ij} x_j \leq u_j$. That follows from (22) and from the definition of γ_{ij} .
 - Constraints (13): $z_{ijk} = \gamma_{ij} t_{ik} x_j = \gamma_{ij} x_j x_i / \bar{u}_i = y_{ij} / \bar{u}_i \leq x_j u_i / \bar{u}_i \leq x_j$ by constraints (23), and since $u_i / \bar{u}_i \leq 1$.
 - Constraints (14): $z_{ijk} = \gamma_{ij} t_{ik} x_j = y_{ij} / \bar{u}_i \geq (u_j x_i + u_i x_j - u_i u_j) / \bar{u}_i \geq t_{ik} u_j + u_i / \bar{u}_i (x_j - u_j) \geq t_{ik} u_j + x_j - u_j$ by constraints (19), and since $u_i / \bar{u}_i \leq 1$, and $x_j - u_j \leq 0$.
2. Let us now prove that for any (x,y,z,t) in \overline{P}_{xyzt} , (x,y) belongs to P_{xy} . Obviously, constraints (18), (19), (20) and (21) are satisfied. We have now to prove that constraints (22) and (23) are satisfied:
 - Constraints (22): by constraints (11), (12) and (16), we have

$$y_{ij} = \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k z_{ijk} \leq \sum_{k=0}^{\lfloor \log(u_i) \rfloor} 2^k u_j t_{ik} = u_j x_i.$$

– Constraints (23): by constraints (18) and (22), we have $y_{ij} = y_{ji} \leq u_i x_j$.

□

Note that constraints (19), (21)-(23) are RLT inequalities [22, 24].

From Theorem 1 and the fact that the objective function of $(\overline{QP}_{\alpha,\beta})$ does not contain the z and t variables, it follows that the polyhedron \overline{P}_{xyzt} in $(\overline{QP}_{\alpha,\beta})$ can be replaced by P_{xy} . More formally, let $(P_{\alpha,\beta})$ be the following program:

$$(P_{\alpha,\beta}) \begin{cases} \text{Min } f_{\alpha,\beta}(x, y) \\ \text{s.t. } (1)(2) \\ x, y \in P_{xy} \end{cases}$$

Corollary 1. *The optimal value of $(\overline{QP}_{\alpha,\beta})$ is equal to the optimal value of $(P_{\alpha,\beta})$.*

3.2 Computing the best convex reformulation

In this section, we show how to compute values of α^* and β^* that make $f_{\alpha^*,\beta^*}(x, y)$ convex, and that maximize the continuous relaxation value of (QP_{α^*,β^*}) . As shown in Corollary 1, this amounts to solving the following problem (CP):

$$(CP) : \begin{aligned} & \max_{\alpha \in \mathbb{R}, \beta \in \mathbf{S}_n} \{v(P_{\alpha,\beta})\} \\ & Q_{\alpha,\beta} \succeq 0 \end{aligned}$$

where $v(P_{\alpha,\beta})$ is the optimal solution value of $(P_{\alpha,\beta})$ and $Q_{\alpha,\beta} = Q + \alpha A^T A + \beta$. We can rewrite (CP) as follows:

$$(CP) : \begin{aligned} & \max_{\alpha \in \mathbb{R}, \beta \in \mathbf{S}_n} \min_{(1)(2)} \{f_{\alpha,\beta}(x, y)\} \\ & Q_{\alpha,\beta} \succeq 0 \quad (x, y) \in P_{xy} \end{aligned}$$

Theorem 2. *Let (SDP) be the following semi-definite program:*

$$(SDP) \begin{cases} \text{Min } f(X, x) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} X_{ij} + \sum_{i=1}^n c_i x_i \\ \text{s.t. } (1)(2) \\ \sum_{r=1}^m \left(\sum_{i=1}^n \left(\sum_{j=1}^n a_{ri} a_{rj} X_{ij} - 2a_{ri} b_r x_i \right) \right) = - \sum_{r=1}^m b_r^2 & (24) \\ X_{ij} \leq u_j x_i & (i, j) \in I^2 \quad (25) \\ X_{ij} \leq u_i x_j & (i, j) \in I^2 \quad (26) \\ -X_{ij} \leq -u_j x_i - u_i x_j + u_i u_j & (i, j) \in I^2 \quad (27) \\ -X_{ij} \leq 0 & (i, j) \in I^2 \quad (28) \\ -X_{ii} \leq -x_i & i \in I \quad (29) \\ \begin{pmatrix} 1 & x \\ x^T & X \end{pmatrix} \succeq 0 & (30) \\ x \in \mathbb{R}^n \quad X \in \mathbf{S}_n & (31) \end{cases}$$

An optimal solution (α^*, β^*) of (CP) can be deduced from the optimal values of the dual variables of (SDP). The optimal coefficient α^* is the optimal value of the dual variable associated with constraint (24). The optimal coefficients β_{ij}^* are computed as $\beta_{ij}^* = \beta_{ij}^{1*} + \beta_{ij}^{2*} - \beta_{ij}^{3*} - \beta_{ij}^{4*}$, for $i \neq j$, and $\beta_{ii}^* = \beta_{ii}^{1*} + \beta_{ii}^{2*} - \beta_{ii}^{3*} - \beta_{ii}^{4*} - \beta_{ii}^{5*}$, where β_{ij}^{1*} , β_{ij}^{2*} , β_{ij}^{3*} , β_{ij}^{4*} , and β_{ii}^{5*} , are the optimal values of the dual variables associated with constraints (25), (26), (27), (28), and (29), respectively.

Proof. We first rewrite $f_{\alpha, \beta}(x, y)$ as follows:

$$f_{\alpha, \beta}(x, y) = \langle Q + \alpha A^T A + \beta, xx^T \rangle + (c - 2\alpha A^T b)^T x - \alpha b^T b + \langle \beta, y \rangle$$

1. Prove that $v(CP) \leq v(SDP)$.

Let $(\bar{\alpha}, \bar{\beta})$ be any feasible solution to (CP), and (\bar{X}, \bar{x}) be any feasible solution to (SDP). The solution (x, y) such that $x = \bar{x}$ and $y = \bar{X}$ is obviously a feasible solution to $(P_{\bar{\alpha}, \bar{\beta}})$. Now, to prove that $v(CP) \leq v(SDP)$, prove that $f_{\bar{\alpha}, \bar{\beta}}(x, y) \leq f(\bar{X}, \bar{x})$, or $f(\bar{X}, \bar{x}) - f_{\bar{\alpha}, \bar{\beta}}(x, y) \geq 0$, i.e that $\langle Q, \bar{X} \rangle - \langle Q + \bar{\alpha} A^T A + \bar{\beta}, \bar{x} \bar{x}^T \rangle + \bar{\alpha} (2(A^T b)^T \bar{x} - b^T b) + \langle \bar{\beta}, \bar{X} \rangle \geq 0$. By constraint (24), we get $2(A^T b)^T \bar{x} - b^T b = \langle A^T A, \bar{X} \rangle$. We have therefore to prove that $\langle Q, \bar{X} \rangle + \bar{\alpha} \langle A^T A, \bar{X} \rangle + \langle \bar{\beta}, \bar{X} \rangle - \langle Q + \bar{\alpha} A^T A + \bar{\beta}, \bar{x} \bar{x}^T \rangle \geq 0$, and thus $\langle Q + \bar{\alpha} A^T A + \bar{\beta}, \bar{X} - \bar{x} \bar{x}^T \rangle \geq 0$. This last inequality is true because $(\bar{\alpha}, \bar{\beta})$ is a feasible solution to (CP), and (\bar{X}, \bar{x}) is a feasible solution to (SDP), and thus both matrices $(Q + \bar{\alpha} A^T A + \bar{\beta})$ and $(\bar{X} - \bar{x} \bar{x}^T)$ belong to \mathbf{S}_n^+ .

2. Prove that $v(CP) \geq v(SDP)$ or equivalently $v(CP) \geq v(DSDP)$ where (DSDP) is the dual of (SDP).

The following problem (DSDP) is the dual of (SDP):

$$(DSDP) \begin{cases} \text{Max } g(\alpha, \beta, \rho, \sigma) = \alpha b^T b - u^T \beta^3 u - \rho^T b - \sigma^T e \\ \text{s.t. } Q + \alpha A^T A + \beta \succeq 0 & (32) \\ c - 2\alpha A^T b - (\beta^1 + \beta^2 - 2\beta^3)^T u + \beta^5 + A^T \rho + D^T \sigma \geq 0 & (33) \\ \beta = \beta^1 + \beta^2 - \beta^3 - \beta^4 - \text{diag}(\beta^5) & (34) \\ \alpha \in \mathbb{R}, \beta \in \mathbf{S}_n, \beta^1, \beta^2, \beta^3, \beta^4 \in \mathbf{S}_n^+, \beta^5 \in \mathbb{R}^n, \rho \in \mathbb{R}^m, \sigma \in \mathbb{R}^p & (35) \end{cases}$$

where ρ and σ are the dual variables associated to constraints (1) and (2), respectively. For $v \in \mathbb{R}^n$, we denote by $M = \text{diag}(v) \in \mathbf{S}_n$ the matrix where $M_{ii} = v_i$ and $M_{ij} = 0 \ \forall i \neq j$.

Let $(\bar{\alpha}, \bar{\beta}, \bar{\rho}, \bar{\sigma})$ be a feasible solution to (DSDP), from constraint (32), we get $Q_{\bar{\alpha}, \bar{\beta}} \succeq 0$ and obviously $(\bar{\alpha}, \bar{\beta})$ is a feasible solution to (CP), which objective value is equal to $v(P_{\bar{\alpha}, \bar{\beta}})$. Let (\bar{x}, \bar{y}) be any feasible solution to $(P_{\bar{\alpha}, \bar{\beta}})$, to prove that $v(CP) \geq v(DSDP)$, we prove that: $f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) \geq g(\bar{\alpha}, \bar{\beta}, \bar{\rho}, \bar{\sigma})$. We have:

$$f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) = \langle Q + \bar{\alpha} A^T A + \bar{\beta}, \bar{x} \bar{x}^T \rangle + (c - 2\bar{\alpha} A^T b)^T \bar{x} + \bar{\alpha} b^T b - \langle \bar{\beta}, \bar{y} \rangle$$

By constraint (34), we get:

$$\begin{aligned} f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) &= \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + (c - 2\bar{\alpha}A^T b)^T \bar{x} + \bar{\alpha}b^T b \\ &\quad - \langle \bar{\beta}^1 + \bar{\beta}^2 - \bar{\beta}^3 - \bar{\beta}^4 - \text{diag}(\bar{\beta}^5), \bar{y} \rangle \\ &= \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + (c - 2\bar{\alpha}A^T b)^T \bar{x} + \bar{\alpha}b^T b \\ &\quad - \langle \bar{\beta}^1, \bar{y} \rangle - \langle \bar{\beta}^2, \bar{y} \rangle + \langle \bar{\beta}^3, \bar{y} \rangle + \langle \bar{\beta}^4, \bar{y} \rangle + \langle \text{diag}(\bar{\beta}^5), \bar{y} \rangle \end{aligned}$$

Moreover, by constraints (19), (20), (21), (22) and (23), and since $\bar{\beta}^1, \bar{\beta}^2, \bar{\beta}^3, \bar{\beta}^4, \bar{\beta}^5, u$ are non-negative, we have:

$$\begin{aligned} f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) &\geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + (c - 2\bar{\alpha}A^T b)^T \bar{x} + \bar{\alpha}b^T b - \langle \bar{\beta}^1, \bar{x}u^T \rangle - \langle \bar{\beta}^2, \bar{x}u^T \rangle \\ &\quad + \langle \bar{\beta}^3, 2\bar{x}u^T - uu^T \rangle + \langle \text{diag}(\bar{\beta}^5), \bar{x} \rangle \\ &\geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + (c - 2\bar{\alpha}A^T b)^T \bar{x} + \bar{\alpha}b^T b - (\bar{\beta}^1 u)^T \bar{x} - (\bar{\beta}^2 u)^T \bar{x} \\ &\quad + 2(\bar{\beta}^3 u)^T \bar{x} - u^T \bar{\beta}^3 u + \bar{\beta}^5 \bar{x}^T \\ &\geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + (c - 2\bar{\alpha}A^T b - (\bar{\beta}^1 + \bar{\beta}^2 - 2\bar{\beta}^3)^T u + \bar{\beta}^5)^T \bar{x} \\ &\quad + \bar{\alpha}b^T b - u^T \bar{\beta}^3 u \end{aligned}$$

By constraints (33), and since x is non-negative, we have :

$$f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) \geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle - (A^T \bar{\rho} + D^T \bar{\sigma})^T \bar{x} + \bar{\alpha}b^T b - u^T \bar{\beta}^3 u$$

Moreover, by constraints (1) and (2)

$$\begin{aligned} f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) &\geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle - \bar{\rho}^T b - \bar{\sigma}^T e + \bar{\alpha}b^T b - u^T \bar{\beta}^3 u \\ &\geq \langle Q + \bar{\alpha}A^T A + \bar{\beta}, \bar{x}\bar{x}^T \rangle + g(\bar{\alpha}, \bar{\beta}, \bar{\rho}, \bar{\sigma}) \end{aligned}$$

Finally, by constraint (32) of $(DSDP)$ we have:

$$f_{\bar{\alpha}, \bar{\beta}}(\bar{x}, \bar{y}) \geq g(\bar{\alpha}, \bar{\beta}, \bar{\rho}, \bar{\sigma})$$

3. First, we proved that from any feasible solution to (SDP) we can design a feasible solution to $(P_{\bar{\alpha}, \bar{\beta}})$, where $(\bar{\alpha}, \bar{\beta})$ is any feasible solution to (CP) , and such that $v(P_{\bar{\alpha}, \bar{\beta}}) \leq v(SDP)$. Hence, that is still true for $(\alpha^{*c}, \beta^{*c})$ an optimal solution to (CP) , and thus we have $v(CP) = v(P_{\alpha^{*c}, \beta^{*c}}) \leq v(SDP)$. Then, we proved that from any feasible solution $(\bar{\alpha}, \bar{\beta}, \bar{\rho}, \bar{\sigma})$ to $(DSDP)$, $(\bar{\alpha}, \bar{\beta})$ is also a feasible solution to (CP) , such that $v(CP) \geq v(DSDP)$. Hence, this is still true for $(\alpha^{*d}, \beta^{*d})$ any optimal solution of $(DSDP)$, and thus we have $v(P_{\alpha^{*d}, \beta^{*d}}) \geq v(DSDP)$. Moreover, since $(\alpha^{*d}, \beta^{*d})$ is a feasible solution to (CP) , and since (CP) is a maximization problem, we have $v(CP) \geq v(P_{\alpha^{*d}, \beta^{*d}}) \geq v(DSDP)$.

To conclude, since $v(SDP) = v(DSDP)$, we deduce that

$$v(CP) \geq v(P_{\alpha^{*d}, \beta^{*d}}) \geq v(DSDP) = v(SDP) \geq v(P_{\alpha^{*c}, \beta^{*c}}) = v(CP),$$

and thus an optimal solution of (CP) is $(\alpha^{*d}, \beta^{*d})$.

□

Remark 1. From the proof of Theorem 2, it follows that for any feasible solution to $(DSDP)$ with an objective value Δ , we can deduce a convex reformulation whose bound by continuous relaxation is not smaller than Δ

4 Dealing with continuous variables

In this section, we adapt our approach to general mixed-integer quadratic programming, following the same reasoning steps as for the pure integer case of Section 3.

A convex Reformulation of (MQP)

We now define the following set of couples of indexes $P = \{(i, j) \in I^2 \cup (I \times J) \cup (J \times I)\}$, and the following perturbed function:

$$h_{\alpha, \beta}(x, y) = h(x) + \sum_{(i, j) \in P} \beta_{ij}(x_i x_j - y_{ij}) + \alpha \sum_{r=1}^m \left(\sum_{i \in I \cup J} a_{ri} x_i - b_r \right)^2,$$

with $\alpha \in \mathbb{R}$, and to simplify the writing, we consider $\beta \in \mathbf{S}_N$, with $\beta_{ij} = 0$, $\forall (i, j) \in J^2$.

Based on the same ideas as in Section 3, we consider the following problem that is equivalent to (MQP):

$$(MQP_{\alpha, \beta}) \begin{cases} \text{Min } h_{\alpha, \beta}(x, y) \\ \text{s.t. } (1)(2) \\ (x, y) \in MP_{xyzt} \end{cases}$$

where MP_{xyzt} is:

$$MP_{xyzt} \begin{cases} (3)(4)(11)(17)(20) \\ (12) - (15) & (i, k) \in E, j \in I \cup J \\ x, y, z, t : (16) & (i, j) \in I \times (I \cup J) \\ (18)(19) & (i, j) \in P \\ (23) & (i, j) \in I \times J \end{cases}$$

Note that the main difference with the pure integer case is that we do not perturb $h(x)$ with $\beta_{ij}(x_i x_j - y_{ij})$ when $(i, j) \in J^2$. The assumption that the subfunction of pure real variables is convex ensures us of the existence of a trivial convex reformulation of (MQP). Indeed, it is always possible to choose α and β in such a way that $h_{\alpha, \beta}(x)$ is convex. For example, take α any non-negative real, $\beta_{ij} = -q_{ij}$, $(i, j) \in (I \times J) \cup (J \times I)$, $\beta_{ij} = 0$, $(i, j) \in I^2$, $i \neq j$, and $\beta_{ii} = -\lambda_{\min}(Q_I)$, $i \in I$ where Q_I is the sub-matrix of Q defined by $(q_{ij})_{(i, j) \in I^2}$. Note that this trivial convex reformulation amounts to convexifying the integer products and to linearizing the mixed integer ones.

Once again, we are interested in the continuous relaxation of $(MQP_{\alpha, \beta})$. Hence, we adapt Theorem 1 to mixed-integer programming.

Theorem 3. *Let \overline{MP}_{xyzt} be the polyhedron obtained by continuous relaxation of MP_{xyzt} (i.e. constraints (17) are replaced by (17')). The projection of \overline{MP}_{xyzt} on variables x and y is the following polyhedron :*

$$MP_{xy} \begin{cases} (4)(20) \\ x, y : (18)(19)(21)(22)(23) \quad (i, j) \in P \end{cases}$$

Proof. Straightforward from the proof of Theorem 1. \square

Computing the best convexification

As in Section 3, we want to compute the values of α and β that maximize the optimal continuous relaxation value of $(MQP_{\alpha,\beta})$. More formally we want to solve:

$$(MCP) : \begin{array}{ll} \max & \min \{h_{\alpha,\beta}(x, y)\} \\ \alpha \in \mathbb{R}, \beta \in \mathbf{S}_N & (1)(2) \\ \beta_{ij}=0, \forall (i,j) \in J^2 & (x,y) \in MP_{xy} \\ Q_{\alpha,\beta} \succeq 0 & \end{array}$$

Theorem 4. *An optimal solution (α^*, β^*) of (MCP) can be deduced from the optimal values of the dual variables of the following semi-definite program:*

$$(SDP') \left\{ \begin{array}{l} \text{Min } f(X, x) = \sum_{i=1}^N \sum_{j=1}^N q_{ij} X_{ij} + \sum_{i=1}^N c_i x_i \\ \text{s.t. } (1)(2)(4)(29)(30) \\ \sum_{r=1}^m \left(\sum_{i=1}^N \left(\sum_{j=1}^N a_{ri} a_{rj} X_{ij} - 2a_{ri} b_r x_i \right) \right) = - \sum_{r=1}^m b_r^2 \quad (24') \\ (25) - (28) \quad (i, j) \in P \\ x \in \mathbb{R}^N \quad X \in \mathbf{S}_N \quad (31') \end{array} \right.$$

The optimal coefficients α^* and β^* are deduced in the same way as in Theorem 2.

Proof. Straightforward from the proof of Theorem 2. \square

Example . Let (MQP_e) be an instance of (MQP) with 2 integer variables and 2 continuous variables:

$$(MQP_e) \left\{ \begin{array}{l} \text{Min } f(x) = x^T \left(\begin{array}{cc|cc} -7 & 3 & -15 & -4 \\ 3 & -14 & -7 & -13 \\ \hline -15 & -7 & 8 & 7 \\ -4 & -13 & 7 & 12 \end{array} \right) x + \begin{pmatrix} 15 \\ 10 \\ -7 \\ -4 \end{pmatrix}^T x \\ \text{s.t. } 5x_1 + x_2 + 8x_3 + 4x_4 \leq 95 \\ 0 \leq x_i \leq 10 \quad i \in \{1, \dots, 4\} \\ x_1, x_2 \in \mathbb{N} \\ x_3, x_4 \in \mathbb{R} \end{array} \right.$$

Observe that the sub-matrix $\begin{pmatrix} 8 & 7 \\ 7 & 12 \end{pmatrix}$ is positive semi-definite.

An optimal solution of (MQP_e) is $x = (8, 10, 2.03, 7.19)$ and its value is -3434.27 .

Since there is no equality constraint there is no parameter α . Thus, we perturb the Q matrix as follows:

$$\begin{pmatrix} -7 + \beta_{11} & 3 + \beta_{12} & -15 + \beta_{13} & -4 + \beta_{14} \\ 3 + \beta_{12} & -14 + \beta_{22} & -7 + \beta_{23} & -13 + \beta_{24} \\ -15 + \beta_{13} & -7 + \beta_{23} & 8 & 7 \\ -4 + \beta_{14} & -13 + \beta_{24} & 7 & 12 \end{pmatrix}$$

and the optimal values of the β parameter are: $\beta_{11} = 12.25$, $\beta_{12} = 0.29$, $\beta_{13} = 8.74$, $\beta_{14} = 0$, $\beta_{22} = 21.17$, $\beta_{23} = 1.84$, and $\beta_{24} = 5.52$.

For the reformulated problem, the optimal value of the continuous relaxation equals -4002.43 . The gap is then of 16.5%

5 Dealing with inequality constraints

Considering a quadratic program containing p inequality constraints, it is easy to replace each inequality constraint by an equality one, by adding p slack variables. To perform this reformulation we distinguish two cases:

- case 1: Consider the pure integer problem (QP) defined in Section 3, where all the coefficients of the inequality constraints are integers or rationals. To perturb the objective function not only with equality constraints, but also with inequality ones, we reformulate (QP) into an equality constrained program (QP') by adding p integer and upper bounded slack variables. Such a problem can be handled as described in Section 3.
- case 2: Consider the mixed-integer problem (MQP) defined in Section 1. To perturb the objective function with inequality constraints, we reformulate it into a mixed-integer quadratic equality constrained program (MQP') by adding p continuous and upper bounded slack variables. Such a problem can be handled as described in Section 4.

Example (Continued). We transform (MQP_e) into an equivalent mixed-integer problem with one equality constraint using the method described above in case 2:

$$(MQP'_e) \left\{ \begin{array}{l} \text{Min } f(x) = x^T \left(\begin{array}{cc|cc} -7 & 3 & -15 & -4 \\ 3 & -14 & -7 & -13 \\ \hline -15 & -7 & 8 & 7 \\ -4 & -13 & 7 & 12 \end{array} \right) x + \begin{pmatrix} 15 \\ 10 \\ -7 \\ -4 \end{pmatrix}^T x \\ \text{s.t. } 5x_1 + x_2 + 8x_3 + 4x_4 + x_5 = 95 \\ 0 \leq x_i \leq 30 \\ 0 \leq x_5 \leq 95 \\ x_1, x_2 \in \mathbb{N} \\ x_3, x_4, x_5 \in \mathbb{R} \end{array} \right. \quad i \in \{1, \dots, 4\}$$

There is now one equality constraint, and thus in addition to the β parameter, we have an α parameter. We perturb the Q matrix as follows:

$$\begin{pmatrix} -7 + \beta_{11} + 25\alpha & 3 + \beta_{12} + 5\alpha & -15 + \beta_{13} + 40\alpha & -4 + \beta_{14} + 20\alpha & \beta_{15} + 5\alpha \\ 3 + \beta_{12} + 5\alpha & -14 + \beta_{22} + \alpha & -7 + \beta_{23} + 8\alpha & -13 + \beta_{24} + 4\alpha & \beta_{25} + \alpha \\ -15 + \beta_{13} + 40\alpha & -7 + \beta_{23} + 8\alpha & 8 + 64\alpha & 7 + 32\alpha & 8\alpha \\ -4 + \beta_{14} + 20\alpha & -13 + \beta_{24} + 4\alpha & 7 + 32\alpha & 12 + 16\alpha & 4\alpha \\ \beta_{15} + 5\alpha & \beta_{25} + \alpha & 8\alpha & 4\alpha & \alpha \end{pmatrix}$$

where the optimal value of the α parameter is 344.08, and the optimal values of the β parameter are: $\beta_{11} = \beta_{13} = \beta_{14} = 0$, $\beta_{12} = -5.55$, $\beta_{15} = -13.67$, $\beta_{22} = 20.56$, $\beta_{23} = 2.10$, $\beta_{24} = 5.12$, and $\beta_{25} = -0.37$.

For the reformulated problem, the optimal value of the continuous relaxation equals -3434.45 . The gap is then of 0.005% .

6 Computational results

In this section, we experiment our approach using the following exact solution algorithm of mixed-integer non-convex quadratic programs (MQP) based on the MIQCR approach:

Algorithm 1 Solution algorithm to (MQP) based on MIQCR

step 1: If necessary, transform each inequality constraint into an equality one.

step 2: Solve the associate semi-definite program (SDP').

step 3: Deduce α^* and β^* .

step 4: Solve the program (MQP_{α^*, β^*}), by a MIQP solver.

(Its continuous relaxation ($\overline{MQP}_{\alpha^, \beta^*}$) is a convex program with an optimal value equal to the optimal value of (SDP'))*

In the rest of the section, we compare MIQCR with the naive convex reformulation NC, concisely described in Section 3. NC is able to solve (MQP) only when J is empty, i.e. to solve (QP). The idea of NC is to subtract the smallest eigenvalue of the Q matrix from its diagonal terms in order to get a positive semi-definite matrix. We obtain the following program:

$$(QP)_{\lambda_{min}} \left\{ \begin{array}{l} \text{Min } x^T Q x + c^T x - \lambda_{min}(Q) \sum_{i=1}^n (x_i^2 - v_i) \\ \text{s.t. } (1)(2) \\ x_i = \sum_{k=0}^{u_i} k r_{ik} \quad i \in I \quad (36) \\ v_i = \sum_{k=0}^{u_i} k^2 r_{ik} \quad i \in I \quad (37) \\ \sum_{k=0}^{u_i} r_{ik} = 1 \quad i \in I \quad (38) \\ r_{ik} \in \{0, 1\} \quad i \in I, k \in \{0, \dots, u_i\} \quad (39) \end{array} \right.$$

$(QP)_{\lambda_{min}}$ is equivalent to (QP), since $x_i^2 = v_i$ in any feasible solution to $(QP)_{\lambda_{min}}$.

Our experiments concern two different problems with general integer variables (i.e. $J = \emptyset$): the Equality Integer Quadratic Problem ($EIQP$) that consists of minimizing a quadratic function subject to one linear equality constraint, and the Inequality Integer Quadratic Problem ($IIQP$) that consists of minimizing a quadratic function subject to one linear inequality constraint.

$$(EIQP) \left\{ \begin{array}{l} \text{Min } x^T Qx + c^T x \\ \text{s.t. } \sum_{i=1}^n a_i x_i = b \\ 0 \leq x_i \leq u_i \quad i \in I \\ x_i \in \mathbb{N} \quad i \in I \end{array} \right.$$

$$(IIQP) \left\{ \begin{array}{l} \text{Min } x^T Qx + c^T x \\ \text{s.t. } \sum_{i=1}^n d_i x_i \leq e \\ 0 \leq x_i \leq u_i \quad i \in I \\ x_i \in \mathbb{N} \quad i \in I \end{array} \right.$$

For both problems, we generate two classes of instances ($(EIQP_1)$ (resp. $(IIQP_1)$) and $(EIQP_2)$ (resp. $(IIQP_2)$). For each class we generate instances with 20, 30, and 40 variables where the coefficients are randomly generated as follows:

$(EIQP_1)$ and $(IIQP_1)$

- The coefficients of Q and c are integers uniformly distributed in the interval $[-100, 100]$. More precisely, for any $i < j$, a number ν is generated in $[-100, 100]$, and then $q_{ij} = q_{ji} = \nu$.
- The a_i and d_i coefficients are integers uniformly distributed in the interval $[1, 50]$.
- $b = 15 * \sum_{i=1}^n a_i$ and $e = 15 * \sum_{i=1}^n d_i$.
- $u_i = 30, i \in I$.

Note that in these instances the solution, $x_i = 15$ for all i , is feasible.

$(EIQP_2)$ and $(IIQP_2)$

- The coefficients of Q and c are randomly generated as for $(EIQP_1)$ and $(IIQP_1)$.
- The a_i and d_i coefficients are integers uniformly distributed in the interval $[1, 100]$.
- $b = 20 * \sum_{i=1}^n a_i$ and $e = 20 * \sum_{i=1}^n d_i$.
- $u_i = 50, i \in I$.

Note that in these instances the solution, $x_i = 20$ for all i , is feasible.

For each problem and for each $n = 20, 30$, or 40 we generate 5 instances obtaining a total of 60 instances.

Our experiments are carried out on a PC with an Intel core 2 duo processor of 2.8 GHz and 2048 MB of RAM using a Linux operating system. We use the

modeling language `ampl` and the solver Cplex version 11 [18] for solving mixed integer quadratic convex programs, and the solver CSDP [5] for solving semi-definite programs.

The results are presented in Tables 1-4, where each row corresponds to one instance.

Legends of the tables:

- *name*: `Problem_n_nb`, where `n` is the number of variables and `nb` is the instance number.
- *opt*: optimal value of the instance. This value is obtained by keeping the MIQCR solution phase running until the end of the Branch and Bound algorithm even if it takes more than 1 hour.
- For both problems (*IIQP*) and (*EIQP*), columns `NC` present the results for the naive convexification, and columns `MIQCR` for the MIQCR algorithm. As there is no inequality constraints in problems (*EIQP*), it is straightforward that in this case, there is no **step 1** in the MIQCR algorithm. For problems (*IIQP*), columns `MIQCR` present the results for the MIQCR algorithm that uses continuous slack variables to transform the inequality constraint into an equality one, as explained in Section 5 (case 2). In our experiments we have chosen to test this more general method. One might get different results by applying the method devoted to the pure integer case (case 1). For problems (*IIQP*), we also tested the MIQCR algorithm without perturbing the objective function with inequality constraints, i.e. without **step 1**. These results are presented in columns `MIQCR'`.
- *ig (initial gap)*: $\left| \frac{opt - l}{opt} \right| * 100$ where l is the optimal value of the continuous relaxation at the root node.
- *time*: CPU time (in seconds) required by the branch-and-bound algorithm. The time limit is fixed to 1 hour.
- *nodes*: number of nodes visited by the branch-and-bound algorithm.
- *fg (final gap)*: $\left| \frac{opt - b}{opt} \right| * 100$ where b is the value of the best lower bound after 1 hour of solution time.

Tables 1 and 2 present the results for (*EIQP₁*) and (*EIQP₂*). For all the instances, we observe that the bound obtained by continuous relaxation in MIQCR is much better than in NC. Indeed, in (*EIQP₁*) the average initial gap over the 15 instances decreases from 14.44% (for NC) to 0.06% (for MIQCR). In (*EIQP₂*) this initial gap decreases from 18.65% to 0.08%. Hence, this average gap is divided by a factor of about 200 and, unsurprisingly, the average number of Branch and Bound nodes is divided by an important factor (about 3000). Moreover, the initial gap associated with MIQCR remains stable since it varies in (*EIQP₁*) from 0% to 0.21% and, in (*EIQP₂*), from 0% to 0.29%. On the contrary, the initial gap associated with NC varies in (*EIQP₁*) between 7.73% and 21.15%

and, in $(EIQP_2)$, between 11.22% and 24.44%. Except for instances of $(EIQP_1)$ with 20 variables and for the instance $EIQP_2_{20_3}$, NC does not find the proved optimal solution within 1 hour of CPU time, while MIQCR solves 14 instances of $(EIQP_1)$ out of 15 and 13 instances of $(EIQP_2)$ out of 15. However, NC can be viewed as a simple heuristic method since the solution provided after one hour of computation is less than 0.2% from the optimal solution (computed by MIQCR). To get optimal values of instances $EIQP_1_{40_1}$, $EIQP_2_{40_2}$, and $EIQP_2_{40_3}$, MIQCR takes 3868s, 5550s and 9476s, respectively. Finally, let us mention the preprocessing time associated with the solution of the semidefinite programs. In $(EIQP_1)$ it takes 180s, 1200s and 5000s on average, for instances of size 20, 30, and 40, respectively. In $(EIQP_2)$ these average preprocessing times are equal to 70s, 600s, and 3500s, respectively. Observe that, as follows from Remark 1 and because the SDP solvers often provide dual feasible solutions as they progress, the solution of semidefinite programs can be stopped after a fixed time. This possibility is interesting for large instances since SDP solvers generally find a good solution very quickly.

Tables 3 and 4 present the results for $(IIQP_1)$ and $(IIQP_2)$. To test the impact of integrating inequality constraints into the convexification process, we solve the 30 instances by the two algorithms MIQCR' and MIQCR described above. MIQCR' is MIQCR without step 1, i.e. without transforming the inequality into equality. Results for these classes of problems reveal a similar trend as for $(EIQP)$ in terms of initial gap, nodes, and computational time for both methods NC and MIQCR'. However, the initial gap obtained by MIQCR' is less stable than for $(EIQP_1)$ and $(EIQP_2)$; in $(IIQP_1)$ it ranges from 0.05% to 1.55% with an average of 0.72%, and in $(IIQP_2)$ from 0.03% to 17.80% with an average of 4.22%. We can clearly see the improvement of the gap value in the results of the MIQCR column, since the average gap in $(IIQP_1)$ (resp. $(IIQP_2)$) decreases from 0.72% (resp. 4.22%) for MIQCR' to 0.12% (resp. 0.06%) for MIQCR. Moreover, the number of nodes in MIQCR is divided by a factor 3 for $(IIQP_1)$ and by a factor 10 for $(IIQP_2)$, in comparison to MIQCR'. MIQCR is able to solve all the instances while MIQCR' solves only 28 instances out of 30. For $(IIQP_1)$, the solutions of the semidefinite programs associated with MIQCR' require 65s, 750s and 4000s on average, for instances of size 20, 30, and 40, respectively. In the case of MIQCR these average computation times are equal to 240s, 1850s and 8000s, respectively. For $(IIQP_2)$, the solutions of the semidefinite programs associated with MIQCR' require 40s, 300s and 1600s on average, for instances of size 20, 30, and 40, respectively. In the case of MIQCR these average computation times are equal to 250s, 1850s and 9000s, respectively.

7 Conclusion and future research

In this paper, we introduce a solution approach for general mixed-integer quadratic programs (MQP) . This approach, that we called MIQCR, is an extension of an earlier convex reformulation of quadratic programs with 0-1 variables known as

QCR. MIQCR handles general integer variables, continuous variables under a few assumptions, and inequalities.

As QCR, MIQCR has two phases: the first phase consists of building a convex reformulation of (MQP) , and, in the second phase, the reformulated problem is submitted to a MIQP solver. To build our reformulation, we propose a general convex reformulation scheme for (MQP) . Then, we focus on the best reformulation within that scheme, from the continuous relaxation bound point of view. We show that this best reformulation can be obtained from the solution of an SDP relaxation of (MQP) . Computational experiments on two types of problems show that MIQCR is able to solve, within one hour of computation, instances having one constraint and up to 40 integer variables. The good quality of these results is due to the tightness of the bound at the root of the Branch and Bound algorithm and to the efficiency of the MIQP solvers.

An interesting additional result is that MIQCR constitutes an improvement of QCR for quadratic 0-1 programs. First, it improves the bound obtained by continuous relaxation, and second, it handles continuous variables and inequalities. A second additional result is that MIQCR can be immediately adapted to quadratic constraints, as it mixes ideas of linearization and that of convexification.

Let us lastly mention the introduction of the naive convexification NC as a basic method for computational comparison purposes. Our results show that, finally, NC can be viewed as a simple-to-implement heuristic.

A future research topic may focus on reducing the solution time of MIQCR. One idea is to take benefit from the polyhedron projection property proved in Theorem 1 to design a specialized Branch and Bound algorithm.

Acknowledgments

We would like to thank the editor and an anonymous referee for helpful comments on the paper.

References

1. Audet, C., Hansen, P., Savard, G.: Essays and Surveys in Global Optimization. GERAD 25th Anniversary Series, Springer, New York, (2005)
2. Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming.* **109**, 55-68 (2007)
3. Billionnet, A., Elloumi, S., Plateau, M.-C.: Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: the QCR method. *Discrete Applied Mathematics.* **157**(6) : 1185-1197 (2009)
4. Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Waechter, A.: An algorithmic framework for convex mixed integer nonlinear programming. *Discrete Optimization.* **5**, 186-204, (2005)
5. Borchers, B.: CSDP, A C Library for Semidefinite Programming. *Optimization Methods and Software.* **11**(1), 613-623 (1999)

6. Caprara, A.: Constrained 0-1 quadratic programming: Basic approaches and extensions. *European Journal of Operational Research*. **187**(3), 1494-1503 (2008)
7. Cui, Y.: Dynamic programming algorithms for the optimal cutting of equal rectangles. *Appl.Math. Model.* **29**, 1040-1053 (2005)
8. Floudas, C.A.: *Deterministic Global Optimization*. Kluwer Academic Publishing, Dordrecht, The Netherlands, (2000)
9. Faye, A., Roupin, F.: Partial Lagrangian relaxation for General Quadratic Programming. *4'OR, A Quarterly Journal of Operations Research*, **5**(1), 75-88, (2007)
10. Frangioni, A., Gentile, C.: Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*. **106**, 225-236 (2006)
11. Fu, H.L., Shiue, L., Cheng, X., Du, D.Z., Kim, J.M.: Quadratic Integer Programming with Application in the Chaotic Mappings of Complete Multipartite Graphs. *J. Optim. Theory Appl.* **110** (3), 545-556 (2001)
12. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A guide to the theory of NP-Completeness*. W.H. Freeman, San Francisco, CA, 1979
13. Glover, F., Woolsey, R.E.: Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program. *Operations Research*. **22**, 180-182 (1974)
14. Glover, F.: Improved linear integer programming formulations of nonlinear integer problems. *Management Sci.* **22**, 455-460 (1975)
15. Hammer, P.L., Rubin, A.A.: Some remarks on quadratic programming with 0-1 variables. *Revue Française d'Informatique et de Recherche Opérationnelle*. **4**(3), 67-79 (1970)
16. Helmberg, C., Rendl, F.: Solving Quadratic (0,1)-Problems by Semidefinite Programs and Cutting Planes. *Mathematical Programming*. **82**, 291-315 (1998)
17. Hua, Z.S., Banerjee, P.: Aggregate line capacity design for PWB assembly systems. *Int. J.Prod. Res.* **38**(11), 2417-2441 (2000)
18. ILOG. *ILOG CPLEX 11.0 Reference Manual*. ILOG CPLEX Division, Gentilly, 2008
19. Körner, F.: A New Bound for the Quadratic Knapsack Problem and Its Use in a Branch and Bound Algorithm. *Optimization*. **17**, 643-648 (1986)
20. Körner, F.: An efficient branch and bound algorithm to solve the quadratic integer programming problem. *Computing*. **30**, 253-260 (1983)
21. Liberti, L., Maculan, N.: *Global Optimization: From Theory to Implementation*, Chapter: Nonconvex Optimization and Its Applications. Springer, New York, (2006)
22. McCormick, G.P.: Computability of global solutions to factorable non-convex programs: Part I - Convex underestimating problems. *Mathematical Programming*. **10**(1), 147-175 (1976)
23. Saxena, A., Bonami, P., Lee, J.: *Disjunctive Cuts for Non-Convex Mixed Integer Quadratically Constrained Programs*. IPCO. Bologna, (2008)
24. Sherali, H.D., Adams, W.P.: A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*. **32**(10), 1274-90 (1986)
25. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*. **99**(3), 563-591 (2004)
26. Tawarmalani, M., Sahinidis, N.V.: *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*. Kluwer Academic Publishing, Dordrecht, The Netherlands, (2002)

Table 1. Solution of ($EIQP_1$)

	opt	NC				MIQCR			
		ig	time (s)	nodes	fg	ig	time (s)	nodes	fg
$EIQP_1_{20.1}$	-5311070	12.89	146	342802	0	0.09	68	2496	0
$EIQP_1_{20.2}$	-5098379	10.98	29	81583	0	0.13	5	16	0
$EIQP_1_{20.3}$	-4554397	10.15	7	19396	0	0.05	7	93	0
$EIQP_1_{20.4}$	-5614860	11.92	43	117812	0	0	0	0	0
$EIQP_1_{20.5}$	-4354396	13.70	1419	1992867	0	0.15	18	321	0
$EIQP_1_{30.1}$	-10210390	13.76	-	7689021	4.94	0.04	21	50	0
$EIQP_1_{30.2}$	-11243370	13.40	-	7916032	3.65	0	4	0	0
$EIQP_1_{30.3}$	-9862120	12.23	-	8133101	3.32	0.04	89	713	0
$EIQP_1_{30.4}$	-10720488	18.23	-	7666601	8.06	0.05	31	56	0
$EIQP_1_{30.5}$	-10835084	19.97	-	7690110	10.16	0.09	179	1217	0
$EIQP_1_{40.1}$	-20907112	7.73	-	6459201	1.31	0.04	-	3961	0.01
$EIQP_1_{40.2}$	-21274411	16.64	-	6037101	10.00	0.04	773	1356	0
$EIQP_1_{40.3}$	-17033610	17.83	-	6022601	10.84	0	157	0	0
$EIQP_1_{40.4}$	-18268074	16.02	-	6119381	9.27	0.03	744	3091	0
$EIQP_1_{40.5}$	-17373411	21.15	-	6064671	14.40	0.21	2337	9526	0

- : Cplex was stopped after 1 hour

Table 2. Solution of ($EIQP_2$)

	opt	NC				MIQCR			
		ig	time (s)	nodes	fg	ig	time (s)	nodes	fg
$EIQP_2_{20.1}$	-9321876	17.84	-	7751701	1.76	0.14	140	3282	0
$EIQP_2_{20.2}$	-9013418	23.11	-	8366287	4.73	0	4	14	0
$EIQP_2_{20.3}$	-15337225	12.53	210	335313	0	0.03	42	926	0
$EIQP_2_{20.4}$	-11863777	17.75	-	8436182	3.48	0.19	86	1393	0
$EIQP_2_{20.5}$	-12095004	21.28	-	8896130	6.53	0.08	14	252	0
$EIQP_2_{30.1}$	-23592535	20.18	-	5832801	11.61	0.29	135	1073	0
$EIQP_2_{30.2}$	-25924713	15.05	-	5700091	5.98	0.11	431	3345	0
$EIQP_2_{30.3}$	-21938906	22.81	-	5814721	13.09	0.02	331	2416	0
$EIQP_2_{30.4}$	-29913305	16.57	-	5754501	7.17	0.05	506	2653	0
$EIQP_2_{30.5}$	-22422891	24.44	-	5968416	14.73	0.12	343	2482	0
$EIQP_2_{40.1}$	-42548497	16.52	-	4474098	10.40	0.02	1162	1315	0
$EIQP_2_{40.2}$	-35957464	24.16	-	4453801	17.04	0.17	-	4236	0.03
$EIQP_2_{40.3}$	-40116963	17.00	-	4526300	10.50	0.05	-	2496	0.03
$EIQP_2_{40.4}$	-51306080	11.22	-	4242061	4.98	0	49	3	0
$EIQP_2_{40.5}$	-38090192	19.31	-	4549921	13.27	0.01	2309	3528	0

- : Cplex was stopped after 1 hour

Table 3. Solution of ($IIQP_1$)

	opt	NC				MIQCR'				MIQCR			
		ig	time (s)	nodes	fg	ig	time (s)	nodes	fg	ig	time (s)	nodes	fg
$IIQP_1_{20_1}$	-4763626	15.28	-	7414712	0.03	1.07	21	342	0	0.48	15	246	0
$IIQP_1_{20_2}$	-3693618	8.22	3	8009	0	0.09	10	69	0	0.06	5	28	0
$IIQP_1_{20_3}$	-3783803	15.10	1557	2560447	0	0.83	11	144	0	0.08	4	20	0
$IIQP_1_{20_4}$	-5624785	7.42	3	8105	0	0.24	4	17	0	0.02	2	0	0
$IIQP_1_{20_5}$	-2749979	10.61	11	32780	0	1.09	11	206	0	0.12	6	54	0
$IIQP_1_{30_1}$	-9104372	4.66	2	3690	0	0.12	32	13	0	0.09	22	22	0
$IIQP_1_{30_2}$	-8632815	12.41	-	7602932	2.69	0.95	174	1192	0	0.05	75	294	0
$IIQP_1_{30_3}$	-7738098	11.73	-	7531701	1.53	0.72	179	996	0	0.01	69	26	0
$IIQP_1_{30_4}$	-6070518	13.53	-	7558301	3.48	1.11	156	867	0	0.27	65	301	0
$IIQP_1_{30_5}$	-10514171	11.94	-	7861657	2.75	0.05	27	30	0	0.05	23	17	0
$IIQP_1_{40_1}$	-14186867	11.43	-	6017301	5.10	0.33	798	1259	0	0.06	296	296	0
$IIQP_1_{40_2}$	-15447872	10.27	-	5456401	3.76	1.55	1015	2886	0	0.03	407	602	0
$IIQP_1_{40_3}$	-11635872	11.79	-	5670701	5.29	0.87	754	2447	0	0.39	607	1041	0
$IIQP_1_{40_4}$	-13820601	11.34	-	5757501	4.78	0.81	389	934	0	0.02	341	472	0
$IIQP_1_{40_5}$	-12820428	11.49	-	5867501	5.43	0.95	799	2810	0	0.13	702	2010	0

- : Cplex was stopped after 1 hour

Table 4. Solution of ($IIQP_2$)

	opt	NC				MIQCR'				MIQCR			
		ig	time (s)	nodes	fg	ig	time (s)	nodes	fg	ig	time (s)	nodes	fg
$IIQP_2_{20_1}$	-11368326	9.62	18	34880	0	0.13	11	100	0	0.02	5	31	0
$IIQP_2_{20_2}$	-8187686	24.73	-	8116218	7.94	8.54	800	33547	0	0.14	23	264	0
$IIQP_2_{20_3}$	-8318158	19.69	-	7916261	3.58	4.49	36	286	0	0.02	12	111	0
$IIQP_2_{20_4}$	-8362534	34.17	-	8178097	16.47	17.80	184	3145	0	0.37	17	189	0
$IIQP_2_{20_5}$	-11277208	10.07	31	58193	0	0.03	7	0	0	0	3	3	0
$IIQP_2_{30_1}$	-28283032	12.21	-	5727121	4.67	0.77	176	450	0	0.05	61	238	0
$IIQP_2_{30_2}$	-24952930	20.73	-	6101801	11.61	2.94	2336	18480	0	0.08	728	3527	0
$IIQP_2_{30_3}$	-20248934	22.67	-	6125254	12.86	8.52	1268	11495	0	0.04	498	2142	0
$IIQP_2_{30_4}$	-24404197	15.14	-	5436644	6.48	2.51	474	2493	0	0.03	43	113	0
$IIQP_2_{30_5}$	-26490741	19.66	-	6073422	10.55	2.68	494	2309	0	0.08	63	379	0
$IIQP_2_{40_1}$	-45892717	14.43	-	4770061	8.25	0.62	676	787	0	0.01	161	81	0
$IIQP_2_{40_2}$	-42436784	20.20	-	5285301	13.52	1.22	3132	2610	0	0	130	100	0
$IIQP_2_{40_3}$	-37091366	24.38	-	5276161	17.49	7.12	-	3259	2.31	0.03	354	642	0
$IIQP_2_{40_4}$	-41852435	20.86	-	5176123	14.35	3.83	-	4258	0.94	0.04	460	490	0
$IIQP_2_{40_5}$	-40834676	18.22	-	5178724	11.92	2.16	2774	5283	0	0.01	544	975	0

- : Cplex was stopped after 1 hour