HAL
open science

# A piecewise-quadratic convexification for exactly solving box-constrained quadratic programs 

Amélie Lambert, Daniel Cosmin Porumbel

## - To cite this version:

Amélie Lambert, Daniel Cosmin Porumbel. A piecewise-quadratic convexification for exactly solving box-constrained quadratic programs. WCGO 2023 - The 2023 World Congress on Global Optimization, Jul 2023, Athens, Greece. hal-04101939

HAL Id: hal-04101939
https://hal.science/hal-04101939
Submitted on 21 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A piecewise-quadratic convexification for exactly solving box-constrained quadratic programs 

Amélie Lambert ${ }^{1}$ and Daniel Porumbel ${ }^{1}$<br>${ }^{1}$ CEDRIC-CNAM, 292 rue saint Martin, 75141 Paris Cedex 03


#### Abstract

This work aims at finding the global minimum of a quadratic function $f$ with box constrained variables. For this goal, we construct a piecewise-quadratic convex relaxation of $f$ defined as the maximum of $k \geq 1$ convex functions. We show that when $k \rightarrow \infty$ the optimal solution of this relaxation converges to an optimal solution of the SDP relaxation of the initial problem. The resulting convexification is tighter that the one produced by previous related methods that use $k=1$. Its integration into a spatial branch-and-bound algorithm brings a second advantage : compared to previous related methods, it can refine the bound at each node by computing new convex functions specifically tailored to act on the considered node. Numerical results suggest that our algorithm can outperform different solvers or other methods based on quadratic convex relaxations.


Keywords Quadratic Programming, piecewisequadratic underestimator, cutting-quadrics

## 1 Introduction

We consider a non-convex Quadratic Boxconstrained Program (QBP) of the form :

$$
(Q B P)\left\{\begin{align*}
& \min f(x) \equiv\left\langle Q, x x^{\top}\right\rangle+c^{\top} x  \tag{1}\\
& \ell_{i} \leq x_{i} \leq u_{i} \quad \forall i \in \mathcal{I}
\end{align*}\right.
$$

where $\langle A, B\rangle=\sum_{i=1}^{n} \sum_{j=1}^{n} a_{i j} b_{i j}, \mathcal{I}=\{1, \ldots, n\}$, $(Q, c, \ell, u) \in \mathcal{S}_{n} \times \mathbb{R}^{n} \times \mathbb{R}_{+}^{n} \times \mathbb{R}^{n}$, and $\mathcal{S}_{n}$ is the set of symmetric matrices of order $n$. Without loss of generality, we assume that the box constraints (1) take the form $x_{i} \in[0,1]$. (QBP) is a fundamental $\mathcal{N} \mathcal{P}$-hard global optimization problem [13]. The most classical methods for
solving it include branch-and-bound algorithms that use various convex relaxations to determine lower bounds, e.g., either linear, quadratic convex or semi-definite relaxations (see [9, 7]). For many of these cases, the quadratic function can be expressed in an extended space of variables, introducing new variables $Y_{i j}$ with $i, j \in \mathcal{I}$ that are meant to satisfy $Y_{i j}=x_{i} x_{j}$. This traditional approach was first used for the linear relaxation, leading to a branch-and-bound based on the relaxation of the non-convex equalities $Y_{i j}=x_{i} x_{j}$ (see for instance [15, 19, 21]).

We also express the problem in this extended ( $x, Y$ ) space, but we go beyond the linear relaxation. We propose the following model which is equivalent to $(Q B P)$ for any semidefinite positive (SDP) matrix $S_{0} \succeq 0$.

$$
\begin{align*}
& \min \left\langle S_{0}, x x^{\top}\right\rangle+c^{\top} x+\left\langle Q-S_{0}, Y\right\rangle  \tag{2a}\\
& Y=x x^{\top}  \tag{2b}\\
& \ell_{i} \leq x_{i} \leq u_{i} \quad i \in \mathcal{I}  \tag{2c}\\
& Y \in \mathcal{S}_{n} \tag{2d}
\end{align*}
$$

A well-known SDP relaxation of ( $Q B P$ ) is obtained by lifting $x$ to a symmetric matrix $X=$ $x x^{T}$ and by relaxing the non-convex constraints $X=x x^{T}$ into $X-x x^{T} \succeq 0$. After linking $X$ and $x$ using the McCormick constraints (3a)-(3d), we obtain the model below, also called the "Shor's plus RLT" relaxation of ( $Q B P$ ) [1].

$$
(S D P)\left\{\begin{array}{l}
\min f(X, x) \equiv\langle Q, X\rangle+c^{T} x \\
X_{i j} \leq u_{j} x_{i}+\ell_{i} x_{j}-\ell_{i} u_{j}, i, j \in \mathcal{I} \text { (3a) } \\
X_{i j} \leq \ell_{j} x_{i}+u_{i} x_{j}-u_{i} \ell_{j}, i, j \in \mathcal{I} \text { (3b) } \\
X_{i j} \geq u_{j} x_{i}+u_{i} x_{j}-u_{i} u_{j}, i, j \in \mathcal{I} \text { (3c) } \\
X_{i j} \geq \ell_{j} x_{i}+\ell_{i} x_{j}-\ell_{i} \ell_{j}, \quad i, j \in \mathcal{I} \text { (3d) } \\
\left(\begin{array}{cc}
1 & x^{T} \\
x & X
\end{array}\right) \succeq 0 \\
x \in \mathbb{R}^{n} \quad X \in \mathcal{S}_{n}
\end{array}\right.
$$



FIGURE 1 - Convex function $f_{S_{0}}$ in black may reach its minimum over many solutions of the $(x, Y)$ space (see the long flat segment $[\mathrm{A}, \mathrm{D}]$ ), because $S_{0}$ may have a large null space, often of dimension close to $\frac{n}{2}$. The resulting convexification in light-gray is weaker than the piecewise-quadratic convexification from the striped area (max of $f_{S_{0}}, f_{S_{1}}$ and $f_{S_{2}}$ ) that has the same minimum; this latter convex function reaches its minimum over a segment $[B, C]$ shorter than $[A, D]$.

By solving the above $(S D P)$ in a preprocessing step, the method MIQCR [5] determines a positive semidefinite matrix $S_{0}$ that maximizes the lower bound associated to the following convex relaxation of $2 \mathrm{ab}-(2 \mathrm{~d})$ : minimize $f_{S_{0}}(x, Y)$ over all $(x, Y)$ that satisfy the McCormick constraints [15] (instead of $Y=x x^{T}$ ), where $f_{S_{0}}$ is the quadratic surface (quadric) given by:

$$
\begin{equation*}
f_{S}(x, Y)=\left\langle S, x x^{\top}\right\rangle+c^{\top} x+\langle Q-S, Y\rangle \tag{4}
\end{equation*}
$$

The main idea of our work is to replace this unique function $f_{S_{0}}$ with multiple functions $f_{S_{k}}$, with $k=0,1,2, \ldots$; we then minimize the function $f^{*}(x, Y)=\max _{k} f_{S_{k}}(x, Y)$ over all $(x, Y)$. Since each function $f_{S_{k}}$ is quadratic and convex, $f^{*}$ is a piecewise-quadratic convex understimator (see Figure 1). We generate the functions $f_{k}$ one by one as in a cutting-planes approach; each function $f_{k}$ aims at cutting the current optimal solution $(x, Y)$ by making it sub-optimal.

Figure 1 shows how this approach based on multiple quadratic cuts $(k>1)$ may generate a tighter convexification than MIQCR $(k=1)$, even if it does not improve the bound computed at the root node of the branch-and-bound. The new algorithm can also provide a better bound than MIQCR at any non-root node, because it can integrate cuts specifically generated to penalize solutions associated to any given node. Notice that it would have been impractical for MIQCR to solve $(S D P)$ at each node.

The paper is organized as follows. Section 2 introduces a parameterized family of relaxations of ( $Q B P$ ) that are piecewise-quadratic and convex; we will show that the optimal value of this relaxation reaches the value of $(S D P)$. Section 3 introduces an iterative Cutting Quadrics Algorithm (CQA) that can be seen as an extension of the cutting-planes algorithm that uses quadratic surfaces (or quadrics) instead of hyperplanes. In Section 4, this algorithm is integrated within a branch-and-bound to solve ( $Q B P$ ) to global optimality. Finally, Section 5 presents experimental results on the boxqp instances, suggesting that our new approach is faster than state-of-the-art solvers, and is able to significantly reduce the number of nodes in comparison to the basic MIQCR.

## 2 A family of convex piecewisequadratic relaxations

Given a set $\mathcal{K}=\left\{S_{k} \succeq \mathbf{0}, k=0,1, \ldots, p\right\}$ of SDP matrices, the multi-cut version of (2a)-2d takes the form below, forming a family of equivalent formulations of $(Q B P)$ indexed by $\mathcal{K}$ :
$\left(P_{\mathcal{K}}\right) \begin{cases}\min t & \\ t \geq\left\langle S_{k}, x x^{\top}\right\rangle+c^{\top} x+\left\langle Q-S_{k}, Y\right\rangle, & S_{k} \in \mathcal{K}( \\ Y=x x^{\top} & \quad i \in \mathcal{I} \\ \ell_{i} \leq x_{i} \leq u_{i} & \\ Y \in \mathcal{S}_{n}, t \in \mathbb{R} & \end{cases}$
Like in the mono-cut version (2a)-2d) of the above program, the only non-convexity of $\left(P_{\mathcal{K}}\right)$ comes from Constraints (5b) that we can classically relax using the McCormick envelopes (6c)(6f). We obtain $\left(\bar{P}_{\mathcal{K}}\right)$ a family of convex relaxations of $(Q B P)$ indexed by a set $\mathcal{K}$ as above :
$\left(\bar{P}_{\mathcal{K}}\right)\left\{\begin{array}{lr}\min t & \quad \text { (6a) } \\ t \geq\left\langle S_{k}, x x^{\top}\right\rangle+c^{\top} x+\left\langle Q-S_{k}, Y\right\rangle, S_{k} \in \mathcal{K} \quad \text { (6b) } \\ Y_{i j} \leq u_{j} x_{i}+\ell_{i} x_{j}-\ell_{i} u_{j}, & i, j \in \mathcal{I} \quad \text { (6c) } \\ Y_{i j} \leq \ell_{j} x_{i}+u_{i} x_{j}-u_{i} \ell_{j}, & i, j \in \mathcal{I} \quad(6 \mathrm{~d}) \\ Y_{i j} \geq u_{j} x_{i}+u_{i} x_{j}-u_{i} u_{j}, & i, j \in \mathcal{I} \quad(6 \mathrm{e}) \\ Y_{i j} \geq \ell_{j} x_{i}+\ell_{i} x_{j}-\ell_{i} \ell_{j}, & i, j \in \mathcal{I} \quad \text { (6f) } \\ \ell_{i} \leq x_{i} \leq u_{i}, & i \in \mathcal{I} \quad \text { (6g) } \\ Y \in \mathcal{S}_{n}, t \in \mathbb{R} & \\ & \end{array}\right.$
Clearly, for any set $\mathcal{K}$, the problem $\left(\bar{P}_{\mathcal{K}}\right)$ is a relaxation of $(Q B P)$, since for any solution $\bar{x}$ of (QBP) of value $t$, the solution $\left(\bar{x}, \bar{x} \bar{x}^{\top}, t\right)$ is feasible for $\left(\bar{P}_{\mathcal{K}}\right)$ with a value of $t$. Moreover, since


FIGURE 2 - The Cutting Quadrics Algorithm (CQA) starts from an initial convex function $f_{S_{0}}$ (blue surface) of problem $\left(\bar{P}_{\mathcal{K}_{0}}\right)$ whose optimal solution is $\left(x^{0}, Y^{0}\right)$, see the small blue disk. Then it generates matrix $S_{1}$ to impose penalty $f_{S_{1}}\left(x^{0}, Y^{0}\right)$ on $\left(x^{0}, Y^{0}\right)$, as marked by the black vertical arrow in both figures. Thus, after adding function $f_{S_{1}}$ (red surface) to $\left(\bar{P}_{\mathcal{K}_{0}}\right)$, the optimal solution of $\left(\bar{P}_{\mathcal{K}_{1}}\right)$ moves to $\left(x^{1}, Y^{1}\right)$.
all matrices $S_{k} \in \mathcal{K}$ are SDP, each constraint (6b) define a quadratic convex set, and so, $\left(\bar{P}_{\mathcal{K}}\right)$ is a convex problem. Now, given an integer $p$, we pose the problem $\left(L B_{p}\right)$ of determining the best set of matrices $\mathcal{K}=\left\{S_{0}, S_{1}, \ldots S_{p}\right\}$, in the sense of leading to the tightest lower bound of $(Q B P)$ :

$$
\left(L B_{p}\right)\left\{\max _{S_{0}, S_{1}, \ldots S_{k} \succeq \mathbf{0}} v\left(\bar{P}_{\left\{S_{0}, S_{1}, \ldots S_{k}\right\}}\right)\right.
$$

This connects our work with previous convex relaxations [4, 5, 11]; when we restrict $\left(\overline{P_{\mathcal{K}}}\right)$ to $\mathcal{K}=\left\{S_{0}\right\}$, we obtain the original MIQCR method from [5]. It is proven in [11], that, the optimal solution of $\left(L B_{1}\right)$ can be deduced from the optimal dual solution of $(S D P)$. It is, moreover, proven in [11] that, if strong duality holds for $(S D P)$, the optimal value of $\left(L B_{1}\right)$ equals the optimal value of $(S D P)$. We now state Proposition 1

Proposition $1 v\left(L B_{\infty}\right)=v\left(L B_{p}\right)=\ldots=$ $v\left(L B_{0}\right)=v(S D P)$.

Proof. The last equality (for $\mathcal{K}=\left\{S_{0}\right\}$ ), follows from [11, Theorem 3.1]. Moreover, by construction, we obviously have $v\left(L B_{\infty}\right) \geq v\left(L B_{p}\right) \geq$ $\ldots \geq v\left(L B_{1}\right)=v(S D P)$. We now show $v\left(L B_{\infty}\right) \leq v(S D P)$. Take a feasible solution $(\bar{X}, \bar{x})$ of (SDP) of objective value $f(\bar{X}, \bar{x})=$ $\langle Q, \bar{X}\rangle+c^{T} \bar{x}$. The solution $(Y=\bar{X}, x=\bar{x})$ is feasible for $\left(\overline{P_{\mathcal{K}}}\right)$ with a value of $\max _{k \in \mathcal{K}}\left\langle S_{k}, \overline{x x}^{\top}-\bar{X}\right\rangle+$ $c^{\top} \bar{x}+\langle Q, \bar{X}\rangle \leq f(\bar{X}, \bar{x})$; this was obtained from
$\left\langle S_{k}, \overline{x x}^{\top}-\bar{X}\right\rangle \leq 0$, which is true for any $S_{k} \succeq 0$ given that $\overline{x x}^{\top}-\bar{X} \preceq 0$ by virtue of (3e).

## 3 The cutting-quadrics algorithm

The Cutting Quadrics Algorithm (CQA) starts from an initial relaxation $\left(\bar{P}_{\mathcal{K}_{0}}\right)$ associated to an initial set $\mathcal{K}_{0}$ of SDP matrices (that may contain the matrix determined by MIQCR). At each iteration $k$, CQA calls a convex solver to determine the optimal solution $\left(x^{k}, Y^{k}, t^{k}\right)$ of $\left(\bar{P}_{\mathcal{K}_{k}}\right)$. It then constructs a matrix $S_{k+1} \succeq 0$ such that $f_{S_{k+1}}\left(x^{k}, Y^{k}\right)>t^{k}$ making current solution $\left(x^{k}, Y^{k}, t^{k}\right)$ infeasible for $\left(\bar{P}_{\mathcal{K}_{k+1}}\right)$, i.e., for the new program (6a)-6h) enriched with $f_{S_{k+1}}$. The next iteration has to move to a new solution, as illustrated in Figure 2. We thus generalize the well-known cutting-planes algorithm by generating convex quadratic cuts instead of separating hyperplanes.

We define $S_{k+1}=r^{k} \cdot v_{\max } v_{\text {max }}^{\top}$, where $v_{\max }$ is the eigenvector of $\left(x^{k} x^{k \top}-Y^{k}\right)$ of maximum eigenvalue. In practice, we can also define $S_{k+1}=$ $r^{k} \sum v_{i} v_{i}^{T}$, where the sum is carried out over all eigenvectors $v_{i}$ having a positive eigenvalue.

The overall solution method is summed up in Algorithm 1, and we prove its convergence in Theorem 1 . Note that we start for set $\mathcal{K}_{0}$ that will at least integrate the null matrix $\mathbf{0}_{\mathbf{n}}$, that corresponds to the standard linearization of $f(x)$.

```
Algorithm 1: The Cutting-Quadrics Algorithm
Input : variable bounds \(\ell\) and \(u\), precision parameters \(\epsilon\) and \(\delta\), (optional) initial matrices \(\mathcal{K}_{0}\)
Output: The best lower bound on ( \(L B_{\infty}\) )
\(\mathcal{K}_{0} \leftarrow \mathcal{K}_{0} \cup\left\{\mathbf{0}_{\mathbf{n}}\right\} \quad / / \mathcal{K}_{0}\) may contain the optimal matrix used by MIQCR
\(\left(x^{0}, Y^{0}, t^{0}\right) \leftarrow \operatorname{Solve}\left(\bar{P}_{\mathcal{K}_{0}}\right) \quad / /\) variable \(t^{0}\) is actually the optimum obj. value
\(k \leftarrow 0\)
while \(\left(x^{k} x^{k T}-Y^{k} \npreceq 0\right) \quad / /\) In practice we use \(\lambda_{\max }\left(x^{k} x^{k T}-Y^{k}\right)>\delta\)
do
\(r^{k}=100+k \quad / /\) A penalty parameter; we keep \(r^{k}=100\) in practice
\(v_{\max } \leftarrow\) the eigenvector of \(x_{k}^{*} x_{k}^{* \top}-Y_{k}^{*}\) of maximum eigenvalue
\(S_{k+1} \leftarrow r^{k} \cdot v_{\max } v_{\max }^{\top} \quad / /\) Or \(S_{k+1}=r \sum v_{i} v_{i}^{T}\), where the sum is carried out over all
// the eigenvectors \(v_{i}\) associated to a positive eigenvalue
\(\mathcal{K}_{k+1}=\mathcal{K}_{k} \cup S_{k+1} \quad / /\) add a new quadratic cut associated to \(S_{k+1}\)
\(\left(x^{k+1}, Y^{k+1}, t^{k+1}\right) \leftarrow \operatorname{Solve}\left(\bar{P}_{\mathcal{K}_{k+1}}\right)\)
\(k \leftarrow k+1\)
if \(t^{k}-t^{k-1}<\epsilon\) then
break // Bound progress too small (only in practice)
```

return $t^{k}$ as the optimal solution of $(Q B P)$ if $Y^{k}=x^{k} x^{k \top}$, or as a lower bound otherwise

Theorem 1 When $k \rightarrow \infty$, the value of the solutions $t^{k}$ generated by Algorithm 1 converge to the optimal value of (SDP).

Proof. We use three steps : i) the new quadric added at iteration $k+1$ separates the current optimal solution $\left(x^{k}, Y^{k}, t^{k}\right)$ of iteration $k$, which implies that the sequence of optimal values $t^{k}$ increases monotonically, $i i$ ) the maximum eigenvalue of $x^{k} x^{k \top}-Y^{k}$ converges to 0 , and iii) $t^{k}$ reaches the optimal value of $(S D P)$ when $k \rightarrow \infty$. (i) Let $\left(x^{k}, Y^{k}, t^{k}\right)$ the optimal solution to $\left(\bar{P}_{\mathcal{K}_{k}}\right)$ of iteration $k$ of objective value $t^{k}$. We first prove that the new quadric $f_{S_{k+1}}$ cuts $\left(x^{k}, Y^{k}, t^{k}\right)$, or equivalently that $f_{S_{k+1}}\left(x^{k}, Y^{k}\right)>f_{S_{k}}\left(x^{k}, Y^{k}\right)=t^{k}$. Let $v$ be the eigenvector of $\left(x^{k} x^{k \top}-Y^{k}\right)$ of maximum eigenvalue $\lambda_{\max }$. By definition, we have $S_{k+1}=r^{k+1} \cdot v v^{\top}$ and we obtain :

$$
\begin{aligned}
& f_{S_{k+1}}\left(x^{k}, Y^{k}\right)-f_{S_{k}}\left(x^{k}, Y^{k}\right) \\
& =\left\langle S_{k+1}, x^{k} x^{k \top}\right\rangle+c^{\top} x^{k}+\left\langle Q-S_{k+1}, Y^{k}\right\rangle \\
& \quad-\left\langle S_{k}, x^{k} x^{k \top}\right\rangle-c^{\top} x^{k}-\left\langle Q-S_{k}, Y^{k}\right\rangle \\
& = \\
& =\left\langle S_{k+1}-S_{k}, x^{k} x^{k \top}-Y^{k}\right\rangle \\
& =r^{k+1} \lambda_{\max }-\left\langle S_{k}, x^{k} x^{k \top}-Y^{k}\right\rangle \\
& \geq
\end{aligned}
$$

where for the last inequality we used that $S_{k}$ has the form $S_{k}=r^{k} \cdot u u^{\top}$ (where $u$ is an eigenvector of $x^{k-1} x^{k-1 \top}-Y^{k-1}$ ) coupled with the following
well-known property [17, §3.2] :

$$
\lambda_{\max }=\max _{u \in \mathbb{R}^{n}\|u\|=1}\left\langle u u^{\top}, x^{k} x^{k \top}-Y^{k}\right\rangle
$$

This proves $f_{S_{k+1}}\left(x^{k}, Y^{k}\right)>f_{S_{k}}\left(x^{k}, Y^{k}\right)$, because $r^{k+1}>r^{k}$ as imposed by the first line of the while loop.
(ii) We will now prove that for any $\delta>0$ no matter how small, there exist some $\bar{k}$ such that $\lambda_{\text {max }}\left(x^{k} x^{k \top}-Y^{k}\right)<\delta$ for any $k \geq \bar{k}$. Suppose for the sake of contradiction that this is not the case. This means that the algorithm generates an infinite number of matrices $\left(x^{k} x^{k \top}-Y^{k}\right) \in \mathcal{S}_{n}$ that all satisfy $\lambda_{\max }\left(x^{k} x^{k \top}-Y^{k}\right) \geq \delta$. The BolzanoWeierstrass theorem states that any infinite sequence in a bounded set (recall that ( $x, Y$ ) belong to the unit hypercube) contains a convergent subsequence. This means that there exists a subsequence $\left(k_{i}\right)$ such that $\left(x^{k_{i}} x^{k_{i} \top}-Y^{k_{i}}\right)$ converges to some fixed point $\overline{x x}^{\top}-\bar{Y}$ when $i \rightarrow \infty$. By assumption, this convergence point satisfies $\lambda_{\max }\left(\overline{x x}^{\top}-\bar{Y}\right) \geq \delta$. For any infinitesimal $\epsilon$, there exists some $k_{\epsilon}$ such that for any $k_{i}>k_{\epsilon}$, any element in matrix $\left(x^{k_{i}} x^{k_{i} \top}-Y^{k_{i}}\right)$ is at a distance smaller than $\epsilon$ from the corresponding element of $\left(\overline{x x}^{\top}-\bar{Y}\right)$, i.e., it stays in a neighborhood $N_{\epsilon}$ of $\left(\overline{x x}^{\top}-\bar{Y}\right)$ of $\infty$-norm below $\epsilon$. But here comes the contradiction. For a sufficiently small $\epsilon$, we have $\lambda_{\max }\left(x x^{\top}-Y\right)>\delta^{\prime}>0 \forall(x, Y) \in N_{\epsilon}$ for some $\delta^{\prime}$ arbitrarily close to $\delta$. Thus, for a sufficiently large
$\hat{k}$ and $r^{\hat{k}}=100+k, f_{S_{\hat{k}}}(x, Y)$ includes a penalty of $r^{\hat{k}} \delta^{\prime}$ that is large enough to make any $(x, Y) \in N_{\epsilon}$ sub-optimal. No iteration after $\hat{k}$ can stay in $N_{\epsilon}$. (iii) We now prove that $t^{k} \rightarrow v(S D P)$.

Let us first take any convergence point $(\bar{x}, \bar{Y}, \bar{t})$ of Algorithm 1 that has to satisfy $\left(\overline{x x} \bar{T}^{\top}-\bar{Y}\right) \preceq 0$ and use it to build a feasible solution of (SDP) such that $\bar{t} \geq v(S D P)$. Since $\left(\overline{x x}{ }^{\top}-\bar{Y}\right) \preceq 0$, the solution $(x=\bar{x}, X=\bar{Y})$ is feasible for (SDP) with a value of $c^{\top} \bar{x}+\langle Q, \bar{Y}\rangle \geq v(S D P)$. The optimal value $\bar{t}$ can be written as below (regardless of how exactly $\mathcal{K}$ was constructed) :

$$
\begin{aligned}
\bar{t} & =\max _{S \in \mathcal{K}}\left\langle S, \overline{x x}^{\top}-\bar{Y}\right\rangle+c^{\top} \bar{x}+\langle Q, \bar{Y}\rangle \\
& =c^{\top} \bar{x}+\langle Q, \bar{Y}\rangle \geq v(S D P)
\end{aligned}
$$

We used $\max _{S \in \mathcal{K}}\left\langle S, \overline{x x}^{\top}-\bar{Y}\right\rangle=\left\langle\mathbf{0}_{\mathbf{n}}, \overline{x x}^{\top}-\bar{Y}\right\rangle=$ 0 which is true because $0 \in \mathcal{K}$ and because $\left\langle S, \overline{x x}^{\top}-\bar{Y}\right\rangle \leq 0$ for any SDP matrix $S \in \mathcal{K}$.
This proves that $\bar{t} \geq v(S D P)$ and we now explain why we can not have $\bar{t}>v(S D P)$. Take the optimal solution ( $x, X$ ) of (SDP) and notice that its value in (6a)-(6f) is

$$
\begin{aligned}
& \max _{S \in \mathcal{K}}\left\langle S, x x^{\top}-X\right\rangle+c^{\top} x+\langle Q, X\rangle \\
= & \max _{S \in \mathcal{K}}\left\langle S, x x^{\top}-X\right\rangle+v(S D P) \leq v(S D P),
\end{aligned}
$$

where we simply used $\left\langle S, x x^{\top}-X\right\rangle \leq 0$, which is true by virtue of $S \succeq \mathbf{0}$ and $x x^{\top}-X \preceq \mathbf{0}$.

The proof of Theorem 1 implies the following Corollary.
Corollary 1 Algorithm 1 solves problem (SDP). In particular, if $(\bar{x}, \bar{Y}, \bar{t})$ is a convergence point of Algorithm 1. $(\bar{Y}, \bar{x})$ is an optimal solution to (SDP) with a value of $\bar{t}=v(S D P)$.

Theorem 1 thus state that Algorithm 1 generates the same lower bound as MIQCR, i.e., $\mathrm{v}(S D P)$. However, as mentioned previously, the additional quadrics tighten the convexification by reducing the set of optimal equivalent solutions (like in Figure 22. This increases the potential of the new convexification when integrated in a branch-and-bound algorithm.

Recall Algorithm 1 can start from any set $\mathcal{K}_{0}$. To improve its convergence, we describe hereafter several ways to populate $\mathcal{K}_{0}$. The first one consists in adding to $\mathcal{K}_{0}$ the optimal matrix calculated by MIQCR, which is $S_{0}=Q+\Phi^{1}+\Phi^{2}-$
$\Phi^{3}-\Phi^{4}$, where $\Phi^{1}, \Phi^{2}, \Phi^{3}, \Phi^{4}$ are the symmetric matrices built from the optimal dual variables associated with Constraints (3a)- 3 d ) of $(S D P)$. This approach has the drawback of requiring solving a large SDP problem. A faster way is the following : take the matrix $Q$ and increase its diagonal by the opposite of the smallest eigenvalue of $Q$, i.e., $S_{0}=Q-\lambda_{\min } \cdot I_{n} \succeq 0$. Another choice relies on extracting the convex part of $Q$, by constructing the matrix $S_{0}=\sum \lambda_{i} v_{i} v_{i}^{T}$ where the sum is carried over all non-negative eigenvalues $\lambda_{i}$ of $Q$ associated to eigenvectors $v_{i}$. Preliminary tests show that the latter approach is still very fast and produces better results than the second one. Note moreover that the size of $\mathcal{K}_{0}$ is not limited, so it is possible to include all these matrices in the initial set $\mathcal{K}_{0}$.

## 4 A branch-and-bound algorithm to optimally solve ( $Q B P$ )

A classical way to solve ( $Q B P$ ) is to use a branch-and-bound algorithm (see [3] for a complete description) where the bounding step is based on CQA. We here provide more details on our implementation.

## The variable selection strategy

Let $\epsilon_{b}>0$ be the precision parameter that state if an equality is satisfied, and $\left(x^{k}, Y^{k}\right)$ be the solution of $\left(\bar{P}_{\mathcal{K}_{k}}\right)$ at the current node, two cases are possible:

1. If $-\epsilon_{b} \leq\left(Y^{k}-x^{k} x^{k \top}\right)_{i j} \leq \epsilon_{b}$ for all $i, j \in$ $I$, then $(x, Y)$ is the optimal solution of the considered branch.
2. Else, we first try to determine $i^{*}$ by solving problem below restricted to the diagonal (to $i^{*}=j^{*}$ ). If this fails, we solve the same problem below with no restriction on $i^{*}$ and $j^{*}$ :

$$
\left(i^{*}, j^{*}\right)=\operatorname{argmax}\left|\left(x_{i} x_{j}-Y_{i j}\right)\right|
$$

The branching rules
The feasible interval $\left[\ell_{i^{*}}, u_{i^{*}}\right]$ of the selected variable $x_{i^{*}}$ is always split in half : the two child nodes are $x_{i^{*}} \in\left[\ell_{i^{*}} \frac{\ell_{i^{*}}+u_{i *}}{2}\right]$ and $x_{i^{*}} \in\left[\frac{\ell_{i^{*}}+u_{i^{*}}}{2}, u_{i^{*}}\right]$.

## The node selection strategy

For selecting the next sub-problem we use the
"best-first" selection strategy, i.e., we select the node with the highest evaluated lower bound.

## Upper bounding

We use a rather basic coordinate descent heuristic to determine upper bounds. We start from the solution $x^{k}$ of $\left(\bar{P}_{\mathcal{K}_{k}}\right)$ at the current node (ignoring all $Y^{k}$ components), that is also feasible for the original ( $Q B P$ ). Our coordinate descent begins by iterating over all $i \in[1 . . n]$; for each $i$, we fix all variables to their current values except for $x_{i}^{k}$. We then solve an optimization sub-problem with only one decision variable, namely $x_{i}^{k}$. After scanning all variables once, we can repeat the process as long as there is at least one variable $i$ for which we detect a possible improvement.

## Lower bounding

At each node of the B\&B, we evaluate $\left(\bar{P}_{\mathcal{K}_{k}}\right)$ from scratch, i.e. without inheriting any quadratic cuts produced by CQA at previous nodes. In fact, we only keep the matrices of set $\mathcal{K}_{0} \cup \mathbf{0}_{\mathbf{n}}$ (first step of Algorithm (1). We stop executing CQA on a given node as soon as the progress of the lower bound $t^{k+1}-t^{k}$ becomes smaller than a fixed value $\delta$. We will evaluate two algorithm versions :
CQBB-1 use a value of $\delta=0.01$,
CQBB-2 for this version, we actually force at maximum one iteration per node.
Regarding CQBB-2, notice that to generate a first $S_{1}$ we need a first solution ( $x^{0}, Y^{0}$ ). To accelerate the node evaluation, we proposed a presolving initial iteration to generate an initial solution $\left(x^{0}, Y^{0}\right)$ without fully solving $\left(\bar{P}_{\mathcal{K}_{0}}\right)$. When we first evaluate a new node, we fix all values of $x$ and $Y$ to that of the optimal value of the parent node, with the exception of the variables that no longer satisfy the new McCormiks envelopes. We thus evaluate the resulting simplified problem with fewer variables more rapidly than fully solving $\left(\bar{P}_{\mathcal{K}_{0}}\right)$.

## 5 Numerical results

We evaluate our global optimization algorithm CQBB on 81 purely continuous quadratic instances with box constraints called boxqp [8, 20]. The sizes of the instances vary from $n=20$ to 90 and the densities of matrix $Q$ from $20 \%$, to $100 \%$. An instance with $n$ variables, a density
$d \%$, and whose instance number with the same characteristics is $k$ is named spar-n- $d-k$.

## Experimental environment :

Our experiments were carried out on a server with 2 CPU Intel Xeon of 2.3 GHz , each of them having 32 thread, and $8 * 16 \mathrm{~GB}$ of RAM with a Linux operating system. We will compare to the original MIQCR method [11] and to the standard solvers Cplex 12.9 [14], Baron 21.1.13 [18], and Gurobi 9.1.1 [16].

For both CQBB-1 and CQBB-2, we start with the initial set $\mathcal{K}_{0}=\left\{S_{0}\right\}$, where $S_{0}$ is obtained by solving (SDP) heuristically by calling the solver Mosek [2] together with the Conic Bundle library [12] within a lagrangian duality framework as described in [6]. We only use $r^{k}=100$ to compute all $S_{k}$ with $k \geq 1$, and we solve the convex relaxations ( $\bar{P}_{\mathcal{K}_{k}}$ ) at each CQA iteration using the $C$ interface to the solver Mosek. Finally, we set the total time limit to 1 hour for all methods and the relative optimality gap of the branch-and-bound to $\epsilon_{b}=10^{-4}$. We define $S_{k+1}=r^{k} \sum v_{i} v_{i}^{T}$, summing over all eigenvectors $v_{i}$ associated to an eigenvalue no smaller than 0.01 .

To compare the performances of the solvers, we use a performance profile of the CPU times (see [10]). The basic idea is the following : for each instance $i$ and each solver $s$, we denote by $t_{i s}$ the time for solving instance $i$ by solver $s$, and we define the performance ratio as $r_{i s}=\frac{t_{i s}}{\min _{s}} t_{i s}$.
Let $N$ be the total number of instances considered, an overall assessment of the performance of solver $s$ for a given $\rho$ is given by $P\left(r_{i s} \leq \rho\right)=$ $\frac{1}{N} *$ number of instances $i$ such that $r_{i s} \leq \rho$. In Figure 3. we present the performance profile of the CPU times for all the compared algorithms for the boxqp instances. Our new approach CQBB-2 outperforms the original MIQCR method and compares well with other solvers in terms of the number of instances solved. In fact, Cplex solves 73 instances, Baron 74 solves instances, CQBB-1 solves 77 instances, MIQCR and Gurobi solve 78 instances, and CQBB-2 solves 80 instances, out of 81 within the time limit of 1 hour.

If the performance profile starts for $\rho=1$ with a smaller value $P\left(r_{i s} \leq \rho\right)$ for CQBB-2 than for the commercial solvers, this is because the smallest instances are solved more rapidly by Gurobi


FIGURE 3 - Performance profile of the total CPU time for the boxqp instances with $n=20$ to 90 with a time limit of 1 hour.
or Cplex. However, the focus of our algorithm is on the largest and denser instances; and with regards to them, CQBB-2 dominates the compared methods, by solving more instances in the long run.

Table 1 reports the total number of generated nodes for the boxqp instances with $n \in[30 . .90]$ with a time limit of 1 hour; we also removed the easier instances that require less than 30 nodes for all methods. The number of nodes is significantly reduced for method CQBB-2 in comparison to MIQCR (by a factor of 1.4 on average), and is further reduced with method CQBB-1, i.e. when we allow more CQA iterations per node.

## 6 Conclusions

We have presented a generic approach to solve box-constrained quadratic programs to global optimality. The main idea is to combine the strength of quadratic convex relaxations with the cutting-planes logic. Indeed, instead of considering a unique under-estimator of the objective function, we propose a family of convexifications indexed by an arbitrary number of quadrics.

We have proposed an original algorithm to generate these quadrics one by one. The quadric generated at each iteration is actually a quadratic cut that separates the current optimal solution, acting similarly to a hyper-plane of the well
known cutting-planes method. We proved that this iterative algorithm converges to the optimal value of a tight semidefinite relaxation of (QBP).

To solve (QBP) to global optimality, we have implemented a spatial branch-and-bound algorithm that relies on the bound computed by CQA at each node. Numerical results suggest that the resulting method outperforms standard solvers on the largest instances. Since CQA can refine the bound at each node of the branching tree, we almost systematically reduce the total number of B\&B needed nodes compared to MIQCR.

Many improvements are still possible to make the method reach its full potential, both in theory or in practice. For instance, from a practical point of view, one idea would be to aggregate the matrices of the set $\mathcal{K}$ at each iteration of the algorithm, to solve more rapidly the convex quadratic relaxation. The method can quite easily extend to integrate binary variables, linear or convex constraints, or to minimize the maximum of several non-convex functions (a problem of a different nature, beyond the power of MIQCR).

## References

[1] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. Journal of Global Optimization, 43(2) :471-484, 2009.

| Instance | MIQCR | CQBB 2 | CQBB-1 |
| :---: | :---: | :---: | :---: |
| Spar-030-060-1 | 43 | 31 | 25 |
| Spar-030-070-1 | 85 | 71 | 51 |
| Spar-030-070-3 | 39 | 31 | 9 |
| Spar-030-080-1 | 53 | 69 | 65 |
| Spar-040-040-1 | 511 | 271 | 231 |
| Spar-040-050-1 | 31 | 25 | 25 |
| Spar-040-050-2 | 45 | 35 | 33 |
| Spar-040-060-1 | 299 | 227 | 181 |
| Spar-040-100-3 | 247 | 211 | 181 |
| Spar-050-030-3 | 37 | 11 | 11 |
| Spar-050-040-2 | 35 | 17 | 17 |
| Spar-050-050-2 | 79 | 67 | 61 |
| Spar-050-050-3 | 103 | 133 | 127 |
| Spar-060-020-3 | 65 | 33 | 33 |
| Spar-070-025-1 | 87 | 37 | 37 |
| Spar-070-025-2 | 225 | 127 | 125 |
| Spar-070-025-3 | 243 | 143 | 117 |
| Spar-070-050-1 | 197 | 155 | 147 |
| Spar-070-075-1 | 75 | 63 | 57 |
| Spar-070-075-2 | 1901 | 1551 | 1223 |
| Spar-070-075-3 | 877 | 779 | 727 |
| Spar-080-025-1 | 255 | 425 | 425 |
| Spar-080-025-2 | 857 | 471 | 463 |
| Spar-080-025-3 | 171 | 149 | 123 |
| Spar-080-050-2 | 47 | 77 | 77 |
| Spar-080-050-3 | 293 | 203 | 203 |
| Spar-080-075-1 | 123 | 107 | 105 |
| Spar-080-075-2 | 353 | 335 | 315 |
| Spar-080-075-3 | 1221 | 1031 | 953 |
| Spar-090-025-1 | 3375 | 1707 | 1699 |
| Spar-090-025-2 | 3229 | 1561 | 1547 |
| Spar-090-025-3 | 675 | 405 | 419 |
| Spar-090-050-1 | 1655 | 1473 | 1383 |
| Spar-090-050-3 | 531 | 423 | 429 |
| Spar-090-075-3 | 1319 | 1103 | 981 |

TABLE 1 - Number of nodes per instance for the boxqp instances ( $n=30$ to 90 ).
[2] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. Version 9.2., 2019.
[3] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. Acta Numerica, 22 :1-131, 2013.
[4] 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.
[5] A. Billionnet, S. Elloumi, and A. Lambert. Exact quadratic convex reformulations of mixedinteger quadratically constrained problems. Mathematical Programming, 158(1) :235-266, 2016.
[6] A. Billionnet, S. Elloumi, A. Lambert, and A. Wiegele. Using a Conic Bundle method to
accelerate both phases of a Quadratic Convex Reformulation. INFORMS Journal on Computing, 29(2) :318-331, 2017.
[7] P. Bonami, O. Günlük, and J. Linderoth. Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods. Mathematical Programming Computation, 10(3) :333-382, 2018.
[8] S. Burer and D. Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. Comput Optim Appl, 43 :181-195, 2009.
[9] J. Chen and S. Burer. Globally solving nonconvex quadratic programming problems via completely positive programming. Mathematical Programming Computation, 4(1):33-52, 2012.
[10] D. Dolan and J. Moré. Benchmarking optimization software with performance profiles. Mathematical Programming, 91 :201-213, 1986.
[11] S. Elloumi and A. Lambert. Global solution of non-convex quadratically constrained quadratic programs. Optimization Methods and Software, 34(1) :98-114, 2019.
[12] C. Helmberg. Conic Bundle v0.3.10, 2011.
[13] R. Horst, P. M. Pardalos, and N. V. Thoai. Introduction to global optimization. Kluwer, Dordrecht, 2000.
[14] IBM. IBM CPLEX 12.9 Reference Manual. 'https://www.ibm. com/docs/en/icos/12.9.0?topic= cplex-callable-library-c-api-reference-manual', 2020.
[15] G.P. McCormick. Computability of global solutions to factorable non-convex programs : Part iconvex underestimating problems. Mathematical Programming, 10(1) :147-175, 1976.
[16] GUROBI optimization. GUROBI 9.1, 2021.
[17] Daniel Porumbel. Demystifying the characterization of sdp matrices in mathematical programming, 2022. arXiv prepint nr. 2210.13072.
[18] N.V. Sahinidis and M. Tawarmalani. Baron 9.0.4: Global optimization of mixed-integer nonlinear programs. User's Manual, 2010.
[19] H.D. Sherali and W.P. Adams. A reformulationlinearization technique for solving discrete and continuous nonconvex problems, volume 31. Springer Science \& Business Media, 2013.
[20] D. Vandenbussche and G. Nemhauser. A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. Mathematical Programming, 102(3) :259-275, 2005.
[21] Y. Yajima and T. Fujie. A polyhedral approach for nonconvex quadratic programming problems with box constraints. Journal of Global Optimization, 13(2) :151-170, 1998.

