Cutting planes from the simplex tableau for quadratically constrained optimization problems

Mustafa Vora* and Ashutosh Mahajan†

Indian Institute of Technology Bombay, Mumbai

July 14, 2023

Abstract

We describe a method to generate cutting planes for quadratically constrained optimization problems. The method uses information from the simplex tableau of a linear relaxation of the problem in combination with McCormick estimators. The method is guaranteed to cut off a basic feasible solution of the linear relaxation that violates the quadratic constraints in the problem as long as finite bounds on all variables are available. These cutting planes are computationally cheap, and do not require any special structure in the input problem. The cuts generated by the method are the well-known Reformulation Linearization Technique (RLT) cuts. The procedure produces a large number of violated cuts. Several variants for selecting good cuts are tested. Instead of adding many cuts, one can also add auxiliary variables and a few cuts. Computational testing on benchmark test instances shows that on an average upto 30% of gap from the optimal can be closed.

*email: mustafa.vora@iitb.ac.in
†email: amahajan@iitb.ac.in

1 Introduction

We consider a Quadratically Constrained Optimization (QCO) problem with a single quadratic constraint of the following form

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x^T Q x + a^T x \le d, \\ & Gx = h, \\ & \underline{x} \le x \le \overline{x}, \end{aligned} \tag{QCP1}$$

where $c, a \in \mathbb{R}^n, d \in \mathbb{R}, G \in \mathbb{R}^{k \times n}, h \in \mathbb{R}^k$, and the symmetric matrix $Q \in \mathbb{R}^{n \times n}$ are given as inputs. The cutting plane procedure proposed here assumes one quadratic constraint for notational convenience only. It works for any number of quadratic constraints, by considering them one at a time, as illustrated in the computational experiments in Section 6.

The QCO and its discrete extension, Mixed Integer Quadratically Constrained Optimization (MIQCO), arise in several applications like the pooling problem in petrochemical industry (Misener and Floudas 2009), distillation sequences in chemical plants (Aggarwal and Floudas 1990), wastewater treatment (Ahmetović and Grossmann 2011), trimloss problem in paper industry (Harjunkoski et al. 2001), computational geometry problems (Costa et al. 2013, Kallrath 2009, Audet et al. 2007) and others. QCO and MIQCO are difficult to solve both in theory and practice, especially when the quadratic functions in the constraints are non-convex. Jeroslow (1973) showed that this problem is undecidable when variables are unbounded. When the variables are bounded, algorithms based on branch-and-cut (Belotti et al. 2009, Tawarmalani and Sahinidis 2004) can be used to find solutions that are optimal within some precision.

Given a QCO of the form (QCP1) above, branch-and-cut algorithms require a suitable relaxation. A relaxation should be easy to solve and at the same time be a close approximation to the original problem. A linear relaxation is often used as it is easy to solve repeatedly in a branch-and-cut framework. McCormick (1976) inequalities are commonly used to obtain a linear relaxation of (QCP1). Simplex method is then used to solve this linear relaxation because of two practical reasons. First, simplex method has superior warm starting ability, that is, if a basic solution is known then it is relatively simple to restart the algorithm after the problem is modified, and second, cutting planes can sometimes be derived from the simplex tableau, for example, Gomory Mixed Integer cuts (Gomory 1960), Gomory fractional cuts (Gomory 1958). The procedure proposed here is similar in vein to these two methods. A gist of the method is first provided along with an example, and a detailed description is provided subsequently.

Suppose we have solved a linear relaxation (LP) of (QCP1) using the simplex method and obtained a solution, say x^*, not feasible to the quadratic constraint. The main idea proposed here is to first substitute some or all basic variables in the quadratic constraint using the corresponding row of the simplex tableau. A new quadratic inequality valid for (QCP1) is thus obtained. The substitution ensures that each term in the new quadratic function has at least one nonbasic variable. Each term is then relaxed using McCormick estimators. Since one of the variables in each term is at its bounds, the McCormick estimators are '*tight*' at x^* for the term. The linear inequality obtained as the sum of McCormick estimators will cut off x^*. Here is a toy example to illustrate the procedure.

Example 1.1. Suppose we get the following two rows in the optimal simplex tableau while solving a linear relaxation of a given QCO.

$$\begin{aligned} x_1 + 2x_3 - 3x_4 + 2x_5 &= 0.3, \\ x_2 + x_6 &= 0.5, \\ x_i \in [0,1] \; i &= 1, \ldots, 6. \end{aligned} \tag{1}$$

Here x_3, x_4, x_5, x_6 are nonbasic variables currently at their lower bounds. A basic feasible solution for the relaxation is $x^* = (0.3, 0.5, 0, 0, 0, 0)$. Further suppose that the QCO has a quadratic constraint $x_1 x_2 \leq x_3$ that is not satisfied by x^*. Substitute x_1 in the quadratic constraint using (1) to obtain a new quadratic constraint

$$0.3x_2 - 2x_2x_3 + 3x_2x_4 - 2x_2x_5 \leq x_3 \tag{2}$$

that is valid for the given QCO. We can use term-by-term McCormick underestimators to obtain a relaxation of this new quadratic constraint. That is, we use the inequalities $-2x_3 \leq -2x_2x_3, 0 \leq 3x_2x_4, \text{ and } -2x_5 \leq -2x_2x_5$ to obtain

$$0.3x_2 - 3x_3 - 2x_5 \leq 0.$$

This inequality is valid for the given QCO, and it cuts off x^*.

The rest of the article is outlined as follows. In Section 2 we describe the McCormick estimators and their key properties used in the procedure. In Section 3 we review existing literature. We then describe our procedure in detail in Section 4. We next show some connections of our procedure with Reformulation Linearization Technique (RLT) in Section 5. Finally, in Section 6 we discuss some computational results to show the efficiency of the cuts we generate, and we conclude in Section 7.

Table 1: Under- and over-estimators that are tight at the edges of the box $B = [\underline{x_1}, \overline{x_1}] \times [\underline{x_2}, \overline{x_2}]$ for the function $f(x) = x_1 x_2$

Edge	Underestimator	Overestimator
$x_1 = \underline{x_1}$	$\underline{x_2}x_1 + \underline{x_1}x_2 - \underline{x_1}\,\underline{x_2}$	$\overline{x_2}x_1 + \underline{x_1}x_2 - \underline{x_1}\,\overline{x_2}$
$x_2 = \underline{x_2}$	$\underline{x_2}x_1 + \underline{x_1}x_2 - \underline{x_1}\,\underline{x_2}$	$\underline{x_2}x_1 + \overline{x_1}x_2 - \overline{x_1}\,\underline{x_2}$
$x_1 = \overline{x_1}$	$\overline{x_2}x_1 + \overline{x_1}x_2 - \overline{x_1}\,\overline{x_2}$	$\underline{x_2}x_1 + \overline{x_1}x_2 - \overline{x_1}\,\underline{x_2}$
$x_2 = \overline{x_2}$	$\overline{x_2}x_1 + \overline{x_1}x_2 - \overline{x_1}\,\overline{x_2}$	$\overline{x_2}x_1 + \underline{x_1}x_2 - \underline{x_1}\,\overline{x_2}$

2 Properties of McCormick Estimators

Property P1 - *Under- and over-estimators of a bilinear function:* For a bilinear function $f : \mathbb{R}^n \to \mathbb{R}$ given by $f(x) = x_i x_j$ for some $i, j \in \{1, \ldots, n\}$, over a given box $B = \{x \in \mathbb{R}^n \mid \underline{x} \le x \le \overline{x}\}$ the following inequalities give a pair of underestimators and a pair of overestimators for f over B

$$\left.\begin{array}{l} \underline{x_j}x_i + \underline{x_i}x_j - \underline{x_i}\,\underline{x_j} \\ \overline{x_j}x_i + \overline{x_i}x_j - \overline{x_i}\,\overline{x_j} \end{array}\right\} \le x_i x_j \le \left\{\begin{array}{l} \underline{x_j}x_i + \overline{x_i}x_j - \overline{x_i}\,\underline{x_j} \\ \overline{x_j}x_i + \underline{x_i}x_j - \underline{x_i}\,\overline{x_j} \end{array}\right. \tag{3}$$

These inequalities are the well known McCormick (1976) inequalties for f.

Property P2 - *McCormick inequalities are tight at bounds:* It is well known that when either x_i or x_j is at its bounds (lower or upper), the under- and over-estimators of f are both *tight* i.e. at least one under- and one over-estimator evaluate to function value at that point. The *tight* under- and over-estimators for four different cases (arising from the condition that one of the two variables is at one of its bounds) are given in Table 1. At points when neither variable is at its bounds, there is a gap between the estimators and the function value.

Property P3 - *Under- and over-estimators of a quadratic function:* Given a general quadratic function $f(x) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j$, where $q_{ij} \in \mathbb{R}$ for all $i, j \in \{1, \ldots, n\}$, a linear underestimator or overestimator of f over a given box $B := [\underline{x}, \overline{x}]$ can be obtained using the above McCormick estimators for each term, depending on the sign of q_{ij}. For example, one underestimator of f is

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ q_{ij}>0}}^n q_{ij}(\underline{x_j}x_i + \underline{x_i}x_j - \underline{x_i}\,\underline{x_j}) + \sum_{i=1}^n \sum_{\substack{j=1 \\ q_{ij}<0}}^n q_{ij}(\underline{x_j}x_i + \overline{x_i}x_j - \overline{x_i}\,\underline{x_j}). \tag{4}$$

Note that many underestimators can be obtained by choosing one of the two estimators possible for each term.

Property P4 - *Tight under- and over-estimators of a quadratic function:* Consider a quadratic function f over a box B as described in Property P3 and an $x^* \in \mathbb{R}^n$

such that for every pair (i, j) with $q_{ij} \neq 0$ at least one of x_i, x_j is at one of its bounds. We can find an under- and over-estimator for f that is *tight* at x^* by selecting an appropriate estimator depending on the sign of q_{ij} (using Table 1) for each term in f.

For example, consider the quadratic function $f(x) = x_1^2 - 2x_1x_2 + x_2x_3$ over the box $B = [0, 1]^3$ and let $x^* = (0, 0.5, 1)$. Clearly, for every term in f at least one of the variables is at its bounds at x^*. From Table 1, we can underestimate x_1^2 with $0, -2x_1x_2$ with $-2x_1$, and x_2x_3 with $x_2 + x_3 - 1$ to obtain a *tight* underestimator $-2x_1 + x_2 + x_3 - 1$ of f. Similarly, a *tight* overestimator for f at x^* is $x_1 + x_2$.

Auxiliary variables for McCormick relaxation: A matrix variable $X = xx^T$ is often introduced (Burer and Saxena 2012) to obtain the following reformulation of (QCP1)

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & \langle Q, X \rangle + a^T x \leq d, \\
& Gx = h, \\
& X = xx^T, \\
& \underline{x} \leq x \leq \overline{x},
\end{aligned}
$$

which can then be relaxed using McCormick inequalities described above to obtain the following linear relaxation

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & \langle Q, X \rangle + a^T x \leq d, \\
& Gx = h, \\
& X_{ij} \geq \underline{x_j} x_i + \underline{x_i} x_j - \underline{x_i}\,\underline{x_j}, \\
& X_{ij} \geq \overline{x_j} x_i + \overline{x_i} x_j - \overline{x_i}\,\overline{x_j}, \\
& X_{ij} \leq \underline{x_j} x_i + \overline{x_i} x_j - \overline{x_i}\,\underline{x_j}, \\
& X_{ij} \leq \overline{x_j} x_i + \underline{x_i} x_j - \underline{x_i}\,\overline{x_j}, \\
& \underline{x} \leq x \leq \overline{x}.
\end{aligned}
\tag{5}
$$

3 Literature review

Most state-of-the-art global optimization solvers for nonconvex problems use Branch-and-Bound algorithms augmented by cutting planes, primal heuristics, presolving, infeasibility analysis etc. (Tawarmalani and Sahinidis 2004, Berthold et al. 2012, Belotti et al. 2009, Misener and Floudas 2013). Cutting planes for QCO have also been developed using several approaches both for general purpose QCO and for certain special structures that are commonly seen in applications. Some of these are described below.

Sherali and Adams (1998) described the Reformulation and Linearization Technique (RLT) for product of linear inequalities. This type of linearization has been quite popular and well studied in the literature. Their approach of taking a product of two linear constraints in the problem and then adding auxiliary variables $X_{ij} = x_i x_j$ for product of variables wherever they appear is also used in this work. This approach reduces to McCormick (1976) relaxation for problems with variable bounds as the only linear constraints. Audet et al. (2000) describe a branch-and-cut approach using the RLT method and give four classes of cutting planes derived from RLT. Sherali and Alameddine (1992) describe the RLT approach for bilinear problems. Liberti and Pantelides (2006) describe a graph theoretical algorithm for augmenting relaxation of nonconvex problems using some RLT constraints. Adams and Johnson (1994) give a first order RLT formulation of Quadratic Assignment Problem. Recently, Bestuzheva et al. (2022) give a separation algorithm based on RLT cuts. They identify products of a bound factor and a linear constraint which will not produce a violated inequality. Such products are then discarded, and other products are considered. They also project some linear constraints on a subspace of variables to obtain RLT cuts for a smaller system of inequalities. Luedtke et al. (2012) provide several results on the strength of McCormick relaxations for multilinear problems and show that the McCormick relaxation of a bilinear function is within a constant factor of the convex hull at every point within the bounds of the variables. All these methods try to search for an RLT inequality by trying different combinations of linear and bound constraints. As far as we understand, information from the simplex tableau has not been used earlier to generate RLT inequalities that are guaranteed to cut off a basic feasible solution of the linear relaxation.

Semidefinite programming (SDP) relaxations for QCO are also well studied in the literature. Shor (1987) proposed an SDP relaxation of the QCO by relaxing the constraint $X = xx^T$ to $X - xx^T \succcurlyeq 0$. Saxena et al. (2008) provide a disjunctive approach to generate valid inequalities based on their SDP relaxation. Burer and Saxena (2012) review methods to obtain linear inequalities from SDP.

Given a QCO of the form (QCP1), it can be relaxed by rewriting the matrix Q as a difference of two positive semidefinite matrices. (Bomze 2002, Poljak and Wolkowicz 1995, Zheng et al. 2011). Another related approach is the αBB underestimators developed by Androulakis et al. (1995) and Adjiman et al. (1998).

4 A Procedure for generating cuts

In this section we describe our procedure for the standard form of a linear relaxation of (QCP1). The method is also explained with examples for the inequality form of the linear relaxation in Appendix 8.

Suppose we are given a QCO of the form (QCP1) and its linear relaxation $R = \min\{c^T x \mid Ax = b, \underline{x} \leq x \leq \overline{x}\}$. We assume that all the additional variables, either substituted for quadratic terms or added as slack/surplus variables to obtain the standard form of the relaxation, are included in x, and finite bounds are available for all variables. If R is infeasible, then so is (QCP1), and no cuts are required. Let x^* be the optimal solution of R. If $x^{*T}Qx^* + a^T x^* \leq d$, then x^* is optimal to (QCP1). Otherwise, let B denote the optimal basis matrix identified by the simplex method and N denote the submatrix of A associated with nonbasic variables. The simplex method provides linear equalities of the form $x_B = B^{-1}b - B^{-1}Nx_N$. A cut can be generated as follows. For every term $x_i x_j$ in $x^T Qx$ with nonzero q_{ij}, if both x_i, x_j are basic variables then substitute at least one of the variables with its corresponding simplex row. If one of the two variables is a nonbasic variable, then either substitute the basic variable or leave the term as is. This step ensures that the quadratic function obtained after substitution has at least one nonbasic variable in each term. This substituted quadratic function can then be relaxed using McCormick estimators to obtain a cutting plane. This gives us Algorithm 1 to separate x^* from the feasible region of (QCP1).

Theorem 1. In Algorithm 1, $f(x^*) - d = \pi^T x^* - \pi_0$. Further, the inequality $\pi^T x \leq \pi_0$ is valid for (QCP1) and cuts off x^*.

Proof. Proof: In Algorithm 1, $g(x)$ is a quadratic function obtained by substituting some variables in $f(x)$ by their corresponding rows of simplex tableau, therefore, it is clear that $f(x) = g(x)$ for every point $x \in R$. In particular, $f(x^*) = g(x^*)$. Each quadratic term in $g(x)$ has at least one nonbasic variable. Property (P4) in Section 2 ensures $g(x^*) = \pi^T x^* + k$. Thus, $f(x^*) = \pi^T x^* + k = \pi^T x^* + d - \pi_0$, and hence $\pi^T x^* - \pi_0 = f(x^*) - d > 0$.

Since $f(x) = g(x)$ for x feasible to (QCP1), an underestimator of g is also an underestimator of f for all feasible points of (QCP1). Hence, the inequality $\pi^T x \leq \pi_0$ is valid for (QCP1). □

The cutting planes derived above are computationally cheap since no additional linear programs are solved and no matrix factorizations or eigen values are required. Note that there are several cuts possible for a quadratic constraint. If both variables of a quadratic term are basic, then one can substitute either one of them or both (steps 6, 7 of Algorithm 1). After $g(x)$ is obtained from step 14 of Algorithm 1, it is

Algorithm 1 Cut generating algorithm

Input: A linear relaxation $R = \min\{c^T x \mid Ax = b, \underline{x} \leq x \leq \overline{x}\} \in \mathbb{R}^p$ of a QCO of the form (QCP1), a basic solution x^* with $x^{*T} Q x^* + a^T x^* > d$, set of indices for basic and nonbasic variables B, N respectively, and a row of the optimal simplex tableau for each basic x_i i. e. $x_i + \sum_{j \in N} \alpha_{ij} x_j = \beta_i \ \forall \ i \in B$.

Output: $(\pi, \pi_0) \in \mathbb{R}^{p+1}$ such that $\pi^T x^* > \pi_0$

procedure GenerateCuts
- $f(x) \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$
- $g(x) \leftarrow 0$
- **for** every quadratic term $x_i x_j$ of f, where $q_{ij} \neq 0$ **do**
  - **if** $i, j \in B$ **then**
    - $h(x) \leftarrow q_{ij}(\beta_i - \sum_{k \in N} \alpha_{ik} x_k) x_j$
    - (Optional) substitute x_j by $(\beta_j - \sum_{k \in N} \alpha_{jk} x_k)$ in $h(x)$
  - **else**
    - $h(x) \leftarrow q_{ij} x_i x_j$
    - **if** $i \in B$ **then**
      - (Optional) substitute x_i by $(\beta_i - \sum_{k \in N} \alpha_{ik} x_k)$ in $h(x)$
    - **else if** $j \in B$ **then**
      - (Optional) substitute x_j by $(\beta_j - \sum_{k \in N} \alpha_{jk} x_k)$ in $h(x)$
  - $g(x) \leftarrow g(x) + h(x)$
- **for** every quadratic term $x_i x_j$ of g **do**
  - **if** coefficient of $x_i x_j$ is nonnegative **then**
    - underestimate the term using the appropriate underestimator from Table 1
  - **else**
    - underestimate the term using the appropriate overestimator from Table 1
- Let $\pi^T x + k$ be the linear underestimator obtained. $\pi_0 \leftarrow d - k$

return (π, π_0)

possible that some quadratic terms in $g(x)$ have both the variables at their bounds (for example, during substitution if one substitutes both the basic variables of a quadratic term) and it may happen that both the underestimators for that term can be used for underestimating the term. In that case, we can select either of the estimators or can take a convex combination of the two. Regardless of how one selects the variables or the estimators, the cut violation at x^* is the same. Hence other criteria like sparsity of the cut, range of coefficients etc. maybe needed to pick an appropriate cut. We now give a small example where the cutting plane method converges to the optimal solution in the limit.

Example 4.1. Consider the problem $\min\{x_1 \mid x_1 + 2x_2 = 1, x_2 = x_1^2, 0 \leq x_1, x_2 \leq 1\}$. Let $R = \{x_1 + 2x_2 = 1, 0 \leq x_1, x_2 \leq 1\}$. Optimal solution $(0, \frac{1}{2})$ can be cut off using the McCormick overestimator for the constraint $x_1^2 - x_2 \geq 0,\ x_1 - x_2 \geq 0$. Let us call this iteration - 0. Applying the above procedure after adding a surplus variable, gives us the cut $x_1 - x_2 \geq \frac{2}{11}$ in the original space of variables. Define the sequence $\{b_k\}$ of right hand sides of the cuts added in each iteration e.g. $b_0 = 0, b_1 = \frac{2}{11}$, etc. Now we show that $b_{k+1} > b_k \forall\ k$ and the cuts generated in the k^{th} iteration is of the form $x_1 - x_2 \geq b_k$. Assume this is true for some k, then in $(k+1)^{\text{th}}$ iteration the active constraints will be $x_1 - x_2 \geq b_k, x_1 + 2x_2 = 1$, and therefore the optimal solution $x_{k+1} = (\frac{1+2b_k}{3}, \frac{1-b_k}{3})$. When we apply Algorithm 1 using these as active constraints we get the following cut

$$x_1 - x_2 \geq \frac{11b_k + 2}{4b_k + 11}.$$

Setting $b_{k+1} = \frac{11b_k+2}{4b_k+11}$ and observing that $b_{k+1} > b_k$ whenever $b_k \geq 0$ completes the proof using induction. In the limit $b_k \to \frac{1}{4}$ and $x_k \to (\frac{1}{2}, \frac{1}{4})$, which is optimal solution to the problem.

We do not know whether the method always converges. A pure cutting plane algorithm for general QCO is still an open question. The above example shows that even when it converges, it can be slow.

It is not necessary to use the optimal basis of the relaxation, sometimes using a non-optimal basis or an infeasible basis may result in a better cut as shown below.

Example 4.2. Let P be the problem $\min\{-x_1 - 4x_2 \mid x_1^2 - x_2^2 \geq 3, x_1 + 2x_2 \leq 2, -x_1 + x_2 \leq 2, -2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1\}$. The feasible region of P is shown in Figure 1. Let $R = \min\{-x_1 - 4x_2 \mid x_1 + 2x_2 + s_1 = 2, -x_1 + x_2 + s_2 = 2, -2 \leq x_1 \leq 2, -1 \leq x_2 \leq 1, 0 \leq s_1 \leq 6, 0 \leq s_2 \leq 5\}$ be a relaxation of P. The optimal solution to R is $x^* = (0, 1), s^* = (0, 1)$ with the optimal objective value $z^* = -4$. Algorithm 1 gives the cut $6x_1 + 20x_2 \leq 16$ and the lower bound increases to -3.33.

Instead, consider a sub optimal corner point $\hat{x} = (2, 0), \hat{s} = (0, 4)^T$. Note that the quadratic constraint is already satisfied at this point. Algorithm 1 gives the cut

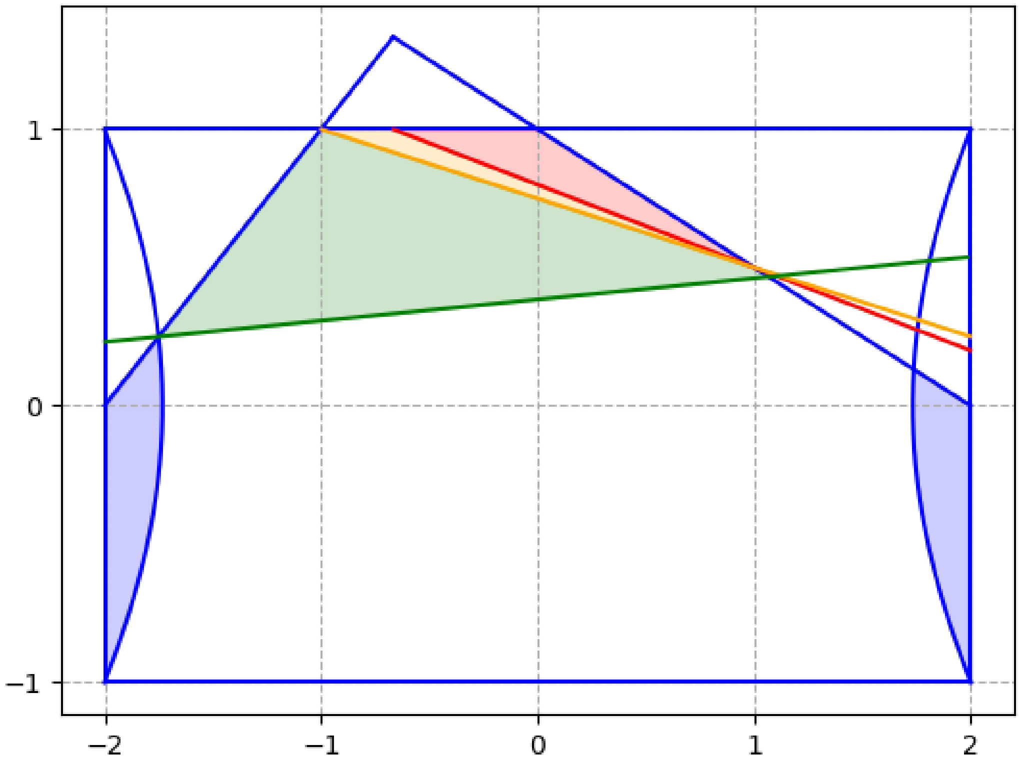

Figure 1: Cuts generated for Example 4.2. Blue region is the feasible region of P, red region is the region cut off from cuts obtained by optimal basis $(x_2, s_1) = (1, 0)$, orange region is the region cut off from cuts obtained by the feasible basis $(x_1, s_1) = (2, 0)$, and green region is the region cut off from cuts obtained by the infeasible basis $(s_1, s_2) = (0, 0)$

$x_1 + 4x_2 \leq 3$, and the lower bound increases to -3. This cut dominates the cut obtained from the optimal basis (see Figure 1). Now choose yet another "corner" point, say, $\hat{x} = (-\frac{2}{3}, \frac{4}{3})^T, \hat{s} = (0,0)^T$ which is infeasible to the relaxation. Algorithm 1 provides the cut $x_1 - 13x_2 \geq -5$ and the lower bound increases to -2.93. This cut dominates the other two cuts obtained.

Thus, carefully choosing a basis to obtain the cuts can impact the performance of Algorithm 1. This observation is not surprising, since, similar techniques have been used for cutting planes for integer linear optimization problems (Cornuéjols 2008).

5 Adding new variables and connections with RLT

In the previous section, linear estimators were obtained for each term of the substituted quadratic function. These estimators were then summed together to obtain a valid inequality. Another way to linearize the substituted quadratic is by adding auxiliary variables for each quadratic term and then using McCormick relaxation as described in Section 2. This procedure obviously creates several new variables that must be added to the LP relaxation. On the other hand, this form is equivalent to adding all the cuts possible after $g(x)$ is obtained by selecting different combinations of under- or over-estimators in Algorithm 1, and hence may tighten the relaxation more. We illustrate this using an example.

Example 5.1. Consider the substituted quadratic inequality (2) obtained in Example 1.1. We add auxiliary variables $w_{23} = x_2x_3, w_{24} = x_2x_4, w_{25} = x_2x_5$ and add

McCormick relaxation for these added variables to obtain the following relaxation

$$\begin{aligned}
x_1 + 2x_3 - 3x_4 + 2x_5 &= 0.3 \\
x_2 + x_6 &= 0.5 \\
0.3x_2 - x_3 - 2w_{23} + 3w_{24} - 2w_{25} &\le 0 \qquad (6)\\
x_2 + x_3 - w_{23} &\le 1 \\
w_{23} - x_2 &\le 0 \\
w_{23} - x_3 &\le 0 \qquad (7)\\
x_2 + x_4 - w_{24} &\le 1 \\
w_{24} - x_2 &\le 0 \\
w_{24} - x_4 &\le 0 \\
x_2 + x_5 - w_{25} &\le 1 \\
w_{25} - x_2 &\le 0 \\
w_{25} - x_5 &\le 0 \qquad (8)\\
x_i &\in [0,1],\ i = 1, \dots, 6 \\
w_{23}, w_{24}, w_{25} &\ge 0
\end{aligned}$$

Taking the linear combination (6) + 2 × ((7) + (8)) and using the fact that $w_{24} \ge 0$ we get the cut $0.3x_2 - 3x_3 - 2x_5 \le 0$ obtained in Example 1.1. Consider the point $\hat{x} = (1, 0.5, 0.15, 1, 1, 0)$ which satisfies both the equality constraints and the cut but there does not exist any $\hat{w}$ such that $(\hat{x}, \hat{w})$ is feasible to the above relaxation. This shows that the above relaxation is tighter than simply adding the cut.

It is clear from the above example that our procedure adds cuts which are equivalent to some of the cuts obtained by Reformulation Linearization Technique (RLT) (Sherali and Adams 1998). However, it should be noted that the substituted quadratic inequality obtained is dense in the nonbasic variables and thus several auxiliary w_{ij} variables will be required. This increases the size of the LP significantly and is practically not suitable for a solver.

6 Computational results

We describe two sets of experiments to assess the computational impact of adding the cuts described in the previous sections. In the first set of experiments (Section 6.1) cuts are added as described in Algorithm 1, i.e., without introducing new variables in the cutting stage. Six variants of this procedure are tested. In the second set of experiments (Section 6.2), new variables are introduced in each round of cutting, as described in Section 5. The second approach results in a much tighter relaxation after adding cuts, but comes with the additional cost of adding more variables each time a cut is added. This experiment is proposed to quantify the effect of deriving one or two inequalities from a quadratic constraint (Section 6.1) relative to adding all possible ones from Algorithm 1.

We implemented the procedures in Minotaur framework (Mahajan et al. 2021). The `mglob` solver in Minotaur solves QCP using branch-and-cut. It is used as a starting point of our implementation. All computational experiments have been performed on a computer with a 64-bit Intel(R) Xeon(R) E5-2670 v2, 2.50GHz CPU, and 128 GB RAM. The programs were run on a single core of the CPU. The code was compiled using GCC-4.9.2 compiler. CLP-1.17.6 (Forrest et al. 2020) was used as an LP solver.

We selected 216 QP, QCP, and QCQP instances from MINLPLib (Bussieck et al. 2003) for the experiments that have an optimal solution available, i.e. there is a gap of less than 10^{-6} between the primal and dual bound in MINLPLib dataset. Limiting our experiments to these "easy" instances enables us to check whether the cuts erroneously cut the optimal point and also to precisely compute the gap closed. We did not consider instances with integer variables as we wanted to focus on our procedure in isolation from other tightening and cut generation procedures. We further excluded 51 instances which either `mglob` solved in the root node without any cuts or the gap between root node relaxation and the optimal objective value was less than 10^{-6}. One instance for which root node processing by Minotaur took more than 30 minutes was also excluded. After this filtering, 164 instances remained for the computational experiments described here. We have chosen both convex and nonconvex problems which contain either quadratic objective or one or more quadratic constraints. Routines to automatically identify and exploit convex nonlinear constraints in `mglob` were disabled for these experiments. Thus all these instances were treated as nonconvex. We call this test set $\mathcal{T}_1$.

We consider another set, $\mathcal{T}_2$ of pooling problems (Misener and Floudas 2009) from the MINLPLib. Pooling problem is a problem in petrochemical industries. All quadratic terms in these instances are bilinear, and may be suited for the RLT cuts like we propose. From a total of 88 pooling instances in MINLPLib dataset, three are not quadratic problems and were removed. One more instance was removed because default Minotaur solves the problem in the root node (without any cuts). Eight instances were removed because default Minotaur takes more than 30 minutes to process the root node. We select the remaining 76 instances. Unlike the set $\mathcal{T}_1$, we do not know the optimal solution value of some of the instances in $\mathcal{T}_2$. Some of these instances have integer or binary variables.

6.1 Cuts in original space of variables

We now describe the computational impact of adding cuts as described in Section 4. The input QCO problem is first transformed by substituting each quadratic term, x_ix_j, that appears in the problem (including the objective function), with an auxiliary variable y_{ij} and then adding the constraint $y_{ij} = x_ix_j$. Bound propagation techniques (Belotti et al. 2010, Puranik and Sahinidis 2017, Domes and Neumaier 2010) are then applied to obtain bounds on each variable. An initial LP relaxation of the transformed problem is then obtained using McCormick inequalities for each bilinear term $y_{ij} = x_ix_j$. We then solve the relaxation to obtain a lower bound that we call z_{before}. Cuts are then added using the proposed variants of Algorithm 1 as described below. The lower bound obtained after adding cutting planes and solving the tightened relaxation is called z_{after}. We compute the gap closed by the cuts using the formula

$$\text{Gap closed} = \frac{(z_{\text{after}} - z_{\text{before}}) \times 100}{z^* - z_{\text{before}}}, \tag{9}$$

where z^* is the best known optimal objective value available from MINLPLib. Note $z^* \geq z_{\text{after}} \geq z_{\text{before}}$.

We have conducted two types of experiments here each having three sub-variants. For each quadratic constraint $y_{ij} = x_ix_j$ when both x_i, x_j are basic variables and $y^*_{ij} \neq x^*_i x^*_j$ (in the initial LP solution), then two possible ways of substituting this quadratic constraint are possible.

1. Substitute both x_i and x_j with their corresponding simplex rows to obtain a quadratic function in only nonbasic variables and then under- or overestimate the new terms to obtain the cuts. We propose three different variants of obtaining the linear estimator of the quadratic function for this case.

   (a) Minimum coefficient sum - Suppose there is a term x_kx_l (after the above substitution) that needs to be overestimated and both x_k, x_l are at their lower bounds $\underline{x_k}, \underline{x_l}$. Then two overestimators $\overline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\overline{x_l}$, and $\underline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\underline{x_l}$ are available. If $|\underline{x_k}| + |\overline{x_l}| < |\overline{x_k}| + |\underline{x_l}|$ then we choose the first overestimator, and the second one otherwise. Similar rules are used for other cases. The motivation behind using the minimum coefficient sum rule is that we prefer smaller coefficients in the cut.

   (b) Equal weight - In this variant, if we have two under- or overestimators for a quadratic term then we take a convex combination of the two estimators with $\lambda = 0.5$.

   (c) Reduced cost weight - Instead of giving equal weights to the two estimators as in (b), reduced costs are used to decide a different convex combination.

Let us consider the term with x_k, x_l and the corresponding reduced costs μ_k, μ_l. If $|\mu_k|+|\mu_l| < \epsilon$ then we select the weights for each estimator as 0.5, otherwise we normalize the reduced costs so that $d_k = \frac{|\mu_k|}{|\mu_k|+|\mu_l|}, d_l = \frac{|\mu_l|}{|\mu_k|+|\mu_l|}$. The underestimators and overestimators then are given in Table 2. We set $\epsilon = 10^{-6}$ in our experiments.

2. In the second set of variants, only one variable is substituted. For a constraint $y_{ij} = x_i x_j$ with $y^*_{ij} \neq x^*_i x^*_j$, we substitute only one out of x_i or x_j with its corresponding simplex row to obtain a new quadratic function. Again, we propose three different ways of choosing which variable to substitute.

   (a) Least infeasible - Among the two variables x_i, x_j we substitute the variable that appears in the fewer number of quadratic constraints that are violated by the current basic solution. If there is a tie we use the one which has fewer nonzero terms in its simplex row.

   (b) Most sparse - Among the two variables x_i, x_j we substitute the variable which has fewer nonzero terms in its simplex row. If there is a tie we use the one that appears in the fewer number of quadratic constraints violated by the current basic solution.

   (c) One-by-one - We substitute both variables one-by-one to obtain two quadratic functions. For example, if the term x_1x_2 needs to substituted and if $x_1 + \sum_{j\in N} \alpha_{1j}x_j = \beta_1$ and $x_2 + \sum_{j\in N} \alpha_{2j}x_j = \beta_2$ are the corresponding simplex rows, then we first substitute x_1 to obtain the quadratic $\beta_1 x_2 - \sum_{j\in N} \alpha_{1j}x_jx_2$ and then we substitute x_2 to obtain another quadratic $\beta_2 x_1 - \sum_{j\in N} \alpha_{2j}x_jx_1$. Thus we get two quadratic functions and two cuts for each quadratic constraint.

We do only a single round of cut generation in these experiments. For all the six variants discussed, we only add a cut to the relaxation if the current LP solution x^* violates it by at least 10^{-3}. In a more practical setting, one would apply these cuts repeatedly and manage them in a more sophisticated manner (see Andreello et al. (2007), Wesselmann and Stuhl (2012), Turner et al. (2022) for example). We leave this aspect of tighter integration with other components of the solver to a future study. We also limit ourselves to only measuring the gap closed by these cuts, and not focus on their overall effectiveness in solving the problems as this would also require a lot of fine tuning and integration with the solver. While performing the experiments upto five instances on some of the variants faced numerical issues, and for such cases we report zero gap closed. The average gap closed per instance in $\mathcal{T}_1$ is tabulated in Table 3. On an average we close about 13% of the gap on the instances

Table 2: Underestimators/overestimators based on the weights from the reduce cost of the variables

State of variable x_k	State of variable x_l	Underestimators to choose	Overestimators to choose
At lower bound ($\underline{x_k}$)	At lower bound ($\underline{x_l}$)	$\underline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\underline{x_l}$	$d_k(\overline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\overline{x_l}) + d_l(\underline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\,\underline{x_l})$
At lower bound ($\underline{x_k}$)	At upper bound ($\overline{x_l}$)	$d_k(\underline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\underline{x_l}) + d_l(\overline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\,\overline{x_l})$	$\overline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\overline{x_l}$
At upper bound ($\overline{x_k}$)	At lower bound ($\underline{x_l}$)	$d_l(\underline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\underline{x_l}) + d_k(\overline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\,\overline{x_l})$	$\underline{x_l}x_k + \overline{x_k}x_l - \underline{x_l}\,\overline{x_k}$
At upper bound ($\overline{x_k}$)	At upper bound ($\overline{x_l}$)	$\overline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\,\overline{x_l}$	$d_l(\overline{x_l}x_k + \underline{x_k}x_l - \underline{x_k}\,\overline{x_l}) + d_k(\underline{x_l}x_k + \overline{x_k}x_l - \overline{x_k}\,\underline{x_l})$

Table 3: Average gap closed after adding the cuts on set $\mathcal{T}_1$.

Substitute both variables	**Minimum Coefficient sum**	**Equal Weight**	**Reduced cost weight**
Average gap closed	12.75	10.45	13.31
Substitute one variable	**Least infeasible**	**Most sparse**	**One-by-one**
Average gap closed	31.06	30.86	35.53

tested when both the basic variables are substituted while more than 30% of gap was closed when only one basic variable is substituted.

We plot a profile in Figure 2, to visualise the distribution of the performance of each of the six variants over 164 instances. The horizontal axis in the plot shows the gap closed (9) and the vertical axis counts the number of instances. A point (x, y) on the plot shows that at least x% gap was closed on y instances. It is clear from the profiles that substituting only one variable is superior to substituting both variables. The choice of sub-variants did not seem to have much influence on the gap closed. The detailed summary of the results for the instances in $\mathcal{T}_1$ for the six variants described here is reported in the file `DatasetT1.csv` in the online supplementary material[1]. We observe that the time taken in cutting is reasonably low for all instances, and that our procedure is computationally cheap. Also, time taken when substituting one variable

[1] https://www.ieor.iitb.ac.in/files/faculty/amahajan/papers/SupplCutPaper.zip

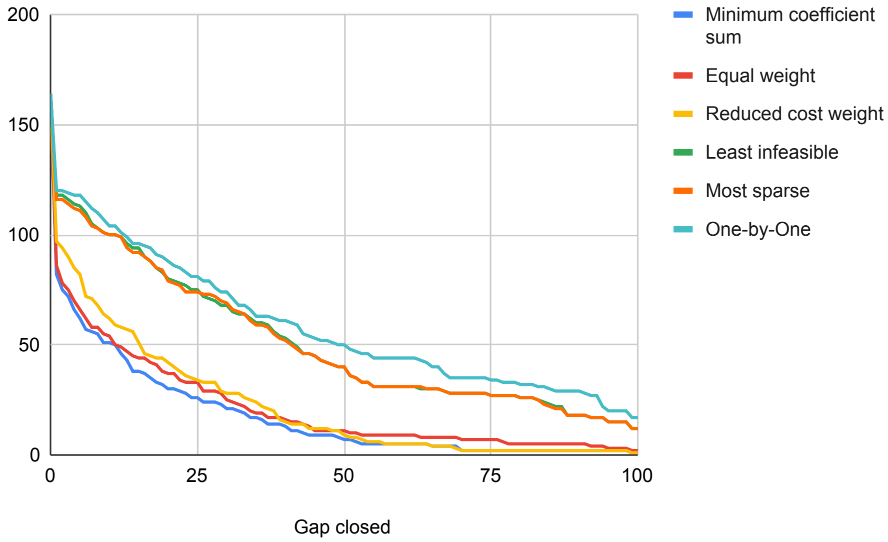

Figure 2: Profile of gap closed by one round of cuts on $\mathcal{T}_1$.

is lower as compared to substituting both variables. It is unsurprising because the number of terms in the new quadratic increases significantly if both variables are substituted.

We also compare the objective lower bound obtained from our cutting plane procedure to that obtained by two different settings of the SCIP 8.0.2. (Bestuzheva et al. 2021). SCIP is one of the leading open-source solvers for integer linear and nonlinear optimization. The first setting keeps the default values of all parameters of SCIP. In the second setting, primal heuristics are turned off in order to isolate the effects of lower bound improvements from cuts and bound tightening routines alone. To switch

Table 4: Comparison of SCIP and Algorithm 1 for $\mathcal{T}_1$ instances

SCIP setting	**Number of instances**	**Number of instances where both SCIP and Algorithm 1 perform similarly**	**Number of instances where Algorithm 1 performs better**	**Number of instances where SCIP performs better**
Default	164	62	27	75
No heuristics	164	50	62	52

Table 5: Comparison of SCIP and Algorithm 1 for $\mathcal{T}_2$ instances

SCIP setting	**Number of instances**	**Number of instances where both SCIP and Algorithm 1 perform similarly**	**Number of instances where Algorithm 1 performs better**	**Number of instances where SCIP performs better**
Default	76	46	13	17
No heuristics	76	43	22	11

off all primal heuristics in SCIP we use the option `set/heuristics/emphasis/off`. We call the lower bound provided by SCIP after processing the root node as z_{SCIP}. Our algorithm is then compared to z_{SCIP} using the formula

$$\text{Percent change} = \frac{(z_{\text{SCIP}} - z_{\text{after}}) \times 100}{|z_{\text{before}}|}, \tag{10}$$

where z_{before} and z_{after} are obtained from Minotaur as described above.

When $|\text{Percent change}| \leq 1$ we say our procedure and SCIP perform similarly. On the other hand if $(\text{Percent change}) < -1$ then we say our procedure performs better than SCIP and if $(\text{Percent change}) > 1$ then we say SCIP performs better than our procedure. SCIP root node processing of some instances in $\mathcal{T}_1$, and $\mathcal{T}_2$ were not completed even after 9600 seconds, and for these instances we use the lower bound provided within this time limit. We summarise the results in Table 4 for test set $\mathcal{T}_1$ and Table 5 for $\mathcal{T}_2$. In both these experiments the variant One-by-One was used as it was the most promising in our previous analysis. We observe that, on an average, the lower bounds from one round of our procedure are inferior to those from default SCIP on $\mathcal{T}_1$ and comparable to those from default SCIP on $\mathcal{T}_2$. The lower bounds from our procedure are seen to be superior to those from SCIP when primal heuristics of SCIP are turned off. It indicates that the proposed procedure may be quite helpful in improving the bounds, and needs good integration with other components of the solver. Detailed summary of the results have been reported in the online supplementary material[2].

[2] https://www.ieor.iitb.ac.in/files/faculty/amahajan/papers/SupplCutPaper.zip

6.2 Adding variables

Now we describe the computational impact when we add auxiliary variables as described in Section 5 to obtain tighter relaxation. We first obtain an initial LP relaxation as explained in Section 6.1. For each quadratic constraint $y_{ij} = x_i x_j$ when both x_i, x_j are basic variables and $y^*_{ij} \neq x^*_i x^*_j$ (in the initial LP solution), we substitute variables with their corresponding simplex row using the following two strategies.

1. Substitute both variables - We substitute both the basic variables x_i and x_j to obtain a new quadratic function. For each term $x_k x_l$ in this new quadratic function, if an auxiliary variable for $x_k x_l$ is already present in the relaxation we substitute the term $x_k x_l$ with that variable. Otherwise we introduce a new variable $w_{kl} = x_k x_l$ and relax it using McCormick relaxations.

2. Substitute one variable - We substitute one variable at a time to obtain two new quadratic functions as described in variant One-by-one in Section 6.1. A new variable w_{kl} is then introduced for each term in the two quadratic constraints as described above.

In both these variants it is sometimes observed that the bounds on w_{kl} introduced can be quite large. If that is the case, the McCormick relaxation can have large coefficients and can cause numerical issues with the LP solvers. If $\max\{|\underline{w_{kl}}|, |\overline{w_{kl}}|\} \geq 10^6$ we do not add a new variable, but rather just add a linear term as described in Section 6.1. This anomaly was observed in 53 instances when both variables were substituted and in 42 instances when one variable was substituted.

We test the above two variants on the test set $\mathcal{T}_1$. Four instances (`torsion*`) hit the time limit of 9600 seconds while generating the new relaxation. These are removed from the subsequent analysis, leaving 160 instances in $\mathcal{T}_1$. Also there were 7 instances facing numerical issues whose gap closed has been reported as 0%. Average gap closed for the two variants is reported in Table 6. We also measure the relative size of the new relaxation in terms of the number of variables in the initial relaxation i.e. the ratio of the number of variables in the new relaxation to the number of variables in the initial relaxation. The second row in Table 6 shows the average relative size of the new relaxation. We observe that substituting both variables and adding auxiliary variables closes 25% of gap on an average in the instances tested while the size of the relaxation increases to more than three times on average. On the other hand substituting one variable at a time and adding auxiliary variables for both quadratic functions closed about 39% of the gap while adding slightly fewer variables. Figure 3 shows the distribution of the performance of both the variants based on the gap closed. The experiment again demonstrates that substituting only one basic variable at a time is more beneficial. Substituting both variables increases the number of terms in the

Table 6: Average gap closed and relative size of the problem after adding auxiliary variables on 160 instances of set $\mathcal{T}_1$

	Substitute both variables	**Substitute one variable**
Average Gap Closed	25.50	39.89
Added varibles w.r.t. original number of variables	3.87	2.84

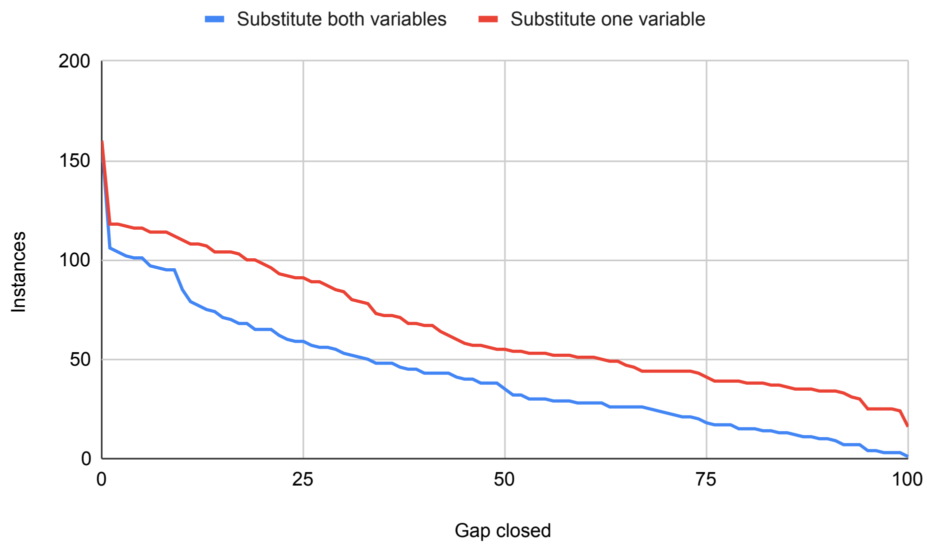

Figure 3: Profile of gap closed by adding auxiliary variables on $\mathcal{T}_1$.

new quadratic whose termwise relaxation can be relatively weak. Detailed summary of results can be accessed in the online supplementary material [3].

The general scheme of introducing new variables while generating cuts is not recommended in a practical setting. Most branch-and-cut implementations do not allow adding new variables after the presolving stage. These experiments however are useful for understanding the effectiveness of the cuts described in Section 6.1. By adding variables, we are introducing all possible cuts that can be generated by Algorithm 1. These experimental results suggest that the heuristic strategy of One-by-one generates sufficiently good cuts and closes a sizeable gap as compared to what is possible by adding all cuts.

[3] https://www.ieor.iitb.ac.in/files/faculty/amahajan/papers/SupplCutPaper.zip

7 Conclusion and Future Work

We have presented a procedure for deriving cutting planes for a linear relaxation of QCP. Our procedure is guaranteed to cut off LP basic feasible solution that is not feasible to the QCP. Our tests of applying one round of cuts yield promising results. Even though these cuts are a particular type of RLT inequalities, they are available readily and do not require any search or guess-work. Successful integration with a general purpose solver would require multiple rounds of cut generation, careful selection, and management of these cuts along with careful tuning of parameters.

There are several open questions with regards to this procedure. First, the convergence of this procedure on general and specific classes of QCP can be analysed. Second, several cuts are possible with different choices available in the algorithm and from different basic solutions. Practical strategies for finding computational effective cuts would be an interesting topic, as would integrating them fully in an MIQCP solver.

References

Warren P Adams and Terri A Johnson. Improved linear programming-based lower bounds for the quadratic assignment problem. *DIMACS series in discrete mathematics and theoretical computer science*, 16:43–77, 1994.

Claire S Adjiman, Stefan Dallwig, Christodoulos A Floudas, and Arnold Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs—I. theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.

A Aggarwal and CA Floudas. Synthesis of general distillation sequences—nonsharp separations. *Computers & chemical engineering*, 14(6):631–653, 1990.

Elvis Ahmetović and Ignacio E Grossmann. Global superstructure optimization for the design of integrated process water networks. *AIChE journal*, 57(2):434–457, 2011.

Giuseppe Andreello, Alberto Caprara, and Matteo Fischetti. Embedding 0, $\frac{1}{2}$-cuts in a branch-and-cut framework: A computational study. *INFORMS Journal on Computing*, 19:229–238, 05 2007.

Ioannis P Androulakis, Costas D Maranas, and Christodoulos A Floudas. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363, 1995.

Charles Audet, Pierre Hansen, Brigitte Jaumard, and Gilles Savard. A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 87(1):131–152, 2000.

Charles Audet, Pierre Hansen, and Frédéric Messine. The small octagon with longest perimeter. *Journal of Combinatorial Theory, Series A*, 114(1):135–150, 2007.

Pietro Belotti, Jon Lee, Leo Liberti, Francois Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software*, 24(4-5):597–634, 2009.

Pietro Belotti, Sonia Cafieri, Jon Lee, and Leo Liberti. Feasibility-based bounds tightening via fixed points. In *International Conference on Combinatorial Optimization and Applications*, pages 65–76. Springer, 2010.

Timo Berthold, Ambros M. Gleixner, Stefan Heinz, and Stefan Vigerske. Analyzing the computational impact of MIQCP solver components. *Numerical Algebra, Control and Optimization*, 2(4):739–748, 2012.

Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 8.0. Technical report, Optimization Online, December 2021.

Ksenia Bestuzheva, Ambros Gleixner, and Tobias Achterberg. Efficient separation of RLT cuts for implicit and explicit bilinear products. *arXiv preprint arXiv:2211.13545*, 2022.

Immanuel M Bomze. Branch-and-bound approaches to standard quadratic optimization problems. *Journal of Global Optimization*, 22(1):17–37, 2002.

Samuel Burer and Anureet Saxena. The MILP road to MIQCP. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 373–405, New York, NY, 2012. Springer New York.

Michael R Bussieck, Arne Stolbjerg Drud, and Alexander Meeraus. MINLPLib—a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.

Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical programming*, 112(1):3–44, 2008.

Alberto Costa, Pierre Hansen, and Leo Liberti. On the impact of symmetry-breaking constraints on spatial branch-and-bound for circle packing in a square. *Discrete Applied Mathematics*, 161(1-2):96–106, 2013.

Ferenc Domes and Arnold Neumaier. Constraint propagation on quadratic constraints. *Constraints*, 15(3):404–429, 2010.

John Forrest, Stefan Vigerske, Ted Ralphs, Lou Hafer, JP Fasano, Haroldo Gambini Santos, Matthew Saltzman, Horand Gassmann, Bjarni Kristjansson, and Alan King. coin-or/Clp: Version 1.17.6 (releases/1.17.6). Zenodo, April 2020.

Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275 – 278, 1958.

Ralph E. Gomory. An algorithm for the mixed Integer Problem. *Report No. P-1885, The Rand Corporation, Santa Monica, CA.*, 1960.

Iiro Harjunkoski, Ray Pörn, and Tapio Westerlund. *MINLP: trim-loss problem*, pages 1469–1477. Springer US, Boston, MA, 2001.

R. C. Jeroslow. There cannot be any algorithm for Integer Programming with Quadratic constraints. *Operations Research*, 21(1):221–224, February 1973.

Josef Kallrath. Cutting circles and polygons from area-minimizing rectangles. *Journal of Global Optimization*, 43(2):299–328, 2009.

Leo Liberti and Constantinos C Pantelides. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, 36(2):161–189, 2006.

James Luedtke, Mahdi Namazifar, and Jeff Linderoth. Some results on the strength of relaxations of multilinear functions. *Mathematical programming*, 136(2):325–351, 2012.

Ashutosh Mahajan, Sven Leyffer, Jeff Linderoth, James Luedtke, and Todd Munson. Minotaur: A mixed-integer nonlinear optimization toolkit. *Mathematical Programming Computation*, 13(2):301–338, 2021.

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part-I Convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Ruth Misener and Christodoulos Floudas. Advances for the pooling problem: Modeling, global optimization, and computational studies survey. *Applied and Computational Mathematics*, 8, 01 2009.

Ruth Misener and Christodoulos A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50, Sep 2013.

Svatopluk Poljak and Henry Wolkowicz. Convex relaxations of (0, 1)-quadratic programming. *Mathematics of Operations Research*, 20(3):550–561, 1995.

Yash Puranik and Nikolaos V Sahinidis. Domain reduction techniques for global NLP and MINLP optimization. *Constraints*, 22(3):338–376, 2017.

Anureet Saxena, Pierre Bonami, and Jon Lee. Disjunctive cuts for non-convex mixed integer quadratically constrained programs. In *Integer Programming and Combinatorial Optimization: 13th International Conference, IPCO 2008 Bertinoro, Italy, May 26-28, 2008 Proceedings 13*, pages 17–33. Springer, 2008.

Hanif D Sherali and WP Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, volume 31. Springer Science & Business Media, 1998.

Hanif D Sherali and Amine Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global optimization*, 2(4):379–410, 1992.

Naum Z Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.

Mohit Tawarmalani and Nikolaos V Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3):563–591, 2004.

Mark Turner, Thorsten Koch, Felipe Serrano, and Michael Winkler. Adaptive cut selection in mixed-integer linear programming. *arXiv preprint arXiv:2202.10962*, 2022.

Franz Wesselmann and U Stuhl. Implementing cutting plane management and selection techniques. *University of Paderborn, Tech. Rep*, 2012.

XJ Zheng, XL Sun, and Duan Li. Nonconvex quadratically constrained quadratic programming: best DC decompositions and their SDP representations. *Journal of Global Optimization*, 50(4):695–712, 2011.

8 APPENDIX : Canonical form of the relaxation

Let $R = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a linear relaxation of the (QCP1). Note that R may have additional auxiliary variables. Let $S = R \cap \{x \in \mathbb{R}^n \mid x^T Qx + a^T x \leq d\}$ be the equivalent feasible region of (QCP1). The inequalities $Ax \leq b$ include lower and upper bound constraints on each variable along with any other additional constraints.

At an optimal extreme point of R, n linearly independent constraints from $Ax \leq b$ will be active. Let such a set of active constraints be $Bx \leq b^B$, where B is a nonsingular square matrix. We can add additional slack variables $s^B \geq 0$ such that $Bx + s^B = b^B$. Since all x feasible to R satisfy $Bx + s^B = b^B$ we get, $x = B^{-1}b^B - B^{-1}s^B$. The optimal solution to R has $x^* = B^{-1}b^B, s^{B*} = 0$. Thus, any feasible solution to R (and also S) must satisfy $x = x^* - B^{-1}s^B$. If x^* is feasible to S then we have obtained an optimal solution to S. Otherwise, the quadratic constraint in S must be violated at x^*, i. e. $x^{*T}Qx^* + a^T x^* > d$.

We substitute $x = x^* - B^{-1}s^B$ on one side of $x^T Qx$ to obtain

$$\begin{aligned} x^T Q(x^* - B^{-1}s^B) + a^T x \leq d \\ \Longrightarrow x^T Qx^* + a^T x - x^T QB^{-1}s^B \leq d. \end{aligned} \tag{11}$$

Let $\widetilde{Q} = \left(\widetilde{q_{ij}}\right) = -QB^{-1}$. Then the quadratic inequality

$$x^T Qx^* + a^T x + \sum_{i=1}^{n} \sum_{j=1}^{n} \widetilde{q_{ij}} x_i s_j^B \leq d, \tag{12}$$

is valid for S. Now we relax this quadratic inequality using McCormick inequalities (3) and Table 1 to get

$$x^TQx^* + a^Tx + \sum_{i=1}^{n}\sum_{\substack{j=1\\ \widetilde{q_{ij}}>0}}^{n} \widetilde{q_{ij}}\underline{x_i}s_j^B + \sum_{i=1}^{n}\sum_{\substack{j=1\\ \widetilde{q_{ij}}<0}}^{n} \widetilde{q_{ij}}\overline{x_i}s_j^B \leq d. \tag{13}$$

At the point $x = x^*, s^{B*} = 0$ the left hand side of the inequality (13) evaluates to $x^{*T}Qx^* + a^Tx^*$. Since we assumed $x^{*T}Qx^* + a^Tx^* > d$ the linear inequality (13) cuts off x^*

Example 8.1. Let $S = \{x \in \mathbb{R}^2 \mid x_1x_2 \leq 4, 4x_1 - 3x_2 \leq 8, 0 \leq x_1, x_2 \leq 4\}$ and $z = \min\{-x_1 \mid x \in S\}$. The optimal $z, z^* = -3$ obtained at $(3, \frac{4}{3})^T$. Consider the linear relaxation $R = \{x \in \mathbb{R}^2 \mid 0 \leq x_1, x_2 \leq 4, 4x_1 - 3x_2 \leq 8\}$. An optimal solution of R is $x^* = (4, \frac{8}{3})^T$, where constraints $x_1 \leq 4$, and $4x_1 - 3x_2 \leq 8$ are active. We have $Q = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 4 & -3 \end{bmatrix}, b_B = \begin{pmatrix} 4 \\ 8 \end{pmatrix}$. Substituting in (12), we obtain the valid quadratic inequality

$$\frac{4}{3}x_1 + 2x_2 - \frac{2}{3}x_1s_1 + \frac{1}{6}x_1s_2^B - \frac{1}{2}x_2s_1^B \leq 4,$$

McCormick underestimators as shown in (13) provide the cut

$$\frac{4}{3}x_1 + 2x_2 - \frac{14}{3}s_1^B \leq 4.$$

Substituting the slack variable using the active constraint we get

$$6x_1 + 2x_2 \leq \frac{68}{3}.$$

Solving the problem after adding the cuts improves the lower bound to $z_l = -3.231$.

In the above procedure when we substitute $x = x^* - B^{-1}s^B$ to obtain (11), we substitute only one of x's in x^TQx, one can substitute both the x's to obtain an inequality in only slack s^B variables, i. e.

$$\begin{aligned} &(x^* - B^{-1}s^B)^TQ(x^* - B^{-1}s^B) + a^T(x^* - B^{-1}s^B) \leq d \\ \Longrightarrow\ & x^{*T}Qx^* + a^Tx^* - 2x^{*T}QB^{-1}s^B - a^TB^{-1}s^B + s^{BT}B^{-T}QB^{-1}s^B \leq d. \end{aligned} \tag{14}$$

Let $\widetilde{Q} = \left(\widetilde{q_{ij}}\right) = B^{-T}QB^{-1}$ then the following quadratic inequality is valid for S

$$x^{*T}Qx^* + a^Tx^* - 2x^{*T}QB^{-1}s^B - a^TB^{-1}s^B + \sum_{i=1}^{n}\sum_{j=1}^{n} \widetilde{q_{ij}}s_i^Bs_j^B \leq d. \tag{15}$$

This quadratic inequality can be underestimated using McCormick underestimators for $\sum_{i=1}^{n}\sum_{j=1}^{n}\widetilde{q_{ij}}s_i^B s_j^B$ to obtain a different cut. In this case we will require to compute the bounds on the s^B variables, which can be computed from the equation $Bx + s^B = b^B$ and bounds on x i. e., $\overline{s^B} = b^B - B_+\underline{x} - B_-\overline{x}$, where B_+ is obtained by replacing all the negative enteries in B with 0 and B_- is obtained by replacing all the positive enteries in B with 0. For every term $\widetilde{q_{ij}}s_i^B s_j^B$ in (12), if $\widetilde{q_{ij}} \geq 0$, then 0 an underestimator for the term and if $\widetilde{q_{ij}} < 0$, then either overestimators $\widetilde{q_{ij}}\overline{s_j^B}s_i^B$ and $\widetilde{q_{ij}}\overline{s_i^B}s_j^B$ can be used (see Table 1 again).

Example 8.2. Consider again the problem from Example 8.1. We substitute Q, B, b^B in (14) to obtain the following quadratic inequality

$$\frac{4}{3}s_1^B s_1^B - \frac{1}{3}s_1^B s_2^B - 8s_1^B + \frac{4}{3}s_2^B \leq -\frac{20}{3}.$$

Note that $s_1^B \in [0, 4], s_2^B \in [0, 20]$. Also, $s_1^B s_1^B$ is underestimated using 0 and $s_1^B s_2^B$ is overestimated using either $20s_1^B$ and $4s_2^B$ to obtain the cuts

$$\begin{aligned} -\frac{44}{3}s_1^B + \frac{4}{3}s_2^B &\leq -\frac{20}{3}, \\ -8s_1^B &\leq -\frac{20}{3}. \end{aligned}$$

Substituting the slack variables and simplifying gives the cuts

$$\begin{aligned} 7x_1 + 3x_2 &\leq 31, \text{ and} \\ x_1 &\leq \frac{19}{6}. \end{aligned}$$

And the lower bound increases to -3.167.