Global QCQP Solver

Chuwen Zhang

May 2, 2021

Contents

1 Semidefinite Relaxation

We consider two types of SDP relaxation for canonical QCQP. We first consider for the case where x is a vector, i.e., $x \in \mathbb{R}^n$.

$$\begin{aligned} \text{Maximize} \quad & x^T Q x \\ \text{s.t.} \quad & x^T A_i x \ (\leq, =, \geq) \ b_i, \forall i = 1, \ldots, m \end{aligned} \tag{1.1}$$

And for inhomogeneous QCQP,

$$\begin{aligned} \text{Maximize} \quad & x^T Q x + 2q^T x \\ \text{s.t.} \quad & x^T A_i x + 2a_i^T x \ (\leq, =, \geq) \ b_i \end{aligned} \tag{1.2}$$

for inhomogeneous case, we notice:

$$\begin{aligned} & x^T Q x + 2q^T x \\ = & \begin{bmatrix} x^T & t \end{bmatrix} \begin{bmatrix} Q & q \\ q^T & o \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} \\ \text{s.t.} \quad & -1 \leq t \leq 1 \end{aligned}$$

we can use a homogeneous reformulation where the size of problem by 1.

1.1 Method I

1.1.1 Vector case

for $X \in \mathbb{R}^n$, we have: $x^T A_i x = A_i \bullet (xx^T)$

$$\begin{aligned} \text{Maximize} \quad & Q \bullet Y \\ \text{s.t.} \quad & Y - xx^T \succeq 0 \text{ or } \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0 \\ & A_i \bullet Y \ (\leq, =, \geq) \ b_i, \forall i \end{aligned} \tag{1.3}$$

1.1.2 Matrix case

for $X \in \mathbb{R}^{n \times d}$, we have: $X^T A_i X = A_i \bullet (XX^T)$

$$\begin{aligned} \text{Maximize} \quad & Q \bullet Y \\ \text{s.t.} \quad & Y - XX^T \succeq 0 \text{ or } \begin{bmatrix} I_d & X^T \\ X & Y \end{bmatrix} \succeq 0 \\ & A_i \bullet Y \ (\leq, =, \geq) \ b_i, \forall i \end{aligned} \tag{1.4}$$

1.1.3 Inhomogeneous

SDP relaxation,

$$\begin{aligned} \text{Maximize} \quad & Q \bullet Y + 2q^T x \\ \text{s.t.} \quad & Y - xx^T \succeq 0 \text{ or } \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0 \\ & A_i \bullet Y + 2a_i^T x \ (\leq, =, \geq) \ b_i, \forall i \end{aligned} \tag{1.5}$$

Alternative

The above formulation could be unbounded. We homogenize by letting $y = (x; t)$,

$$\begin{aligned} \text{Maximize} \quad & y^T \begin{bmatrix} Q & q \\ q^T & 0 \end{bmatrix} y \\ \text{s.t.} \quad & y^T \begin{bmatrix} A_i & a_i \\ a_i^T & 0 \end{bmatrix} y \ (\leq, =, \geq) \ b_i, \forall i \end{aligned} \tag{1.6}$$

1.2 Method II: The Many-Small-Cone

Let $A = A^+ + A_-$ be symmetric where $A_-, A^+ \succeq 0$, which allows Cholesky decomposition,

$$U^+(U^+)^T = A^+$$

since U^+ may be low-rank, we can define z^+ accordingly,

$$\begin{aligned} & (U^+)^T x = z^+ \\ & x^T A^+ x = ||z^+||^2 = \sum_i (z^+)_i^2 \\ & y_i = (z_i^+)^2, \begin{bmatrix} 1 & z_i^+ \\ z_i^+ & y_i \end{bmatrix} \succeq 0, \forall i \end{aligned} \tag{1.7}$$

This is the so-called "many-small-cone" method.

1.2.1 Partition a Matrix into Positive and Negative Parts

We first compute the eigenvalues for $Q, A_i, i = 1, ..., m$ beforehand, then we decompose each matrix by sign of the eigenvalues. One way to do this is:

- Symmetrize: re-define $Q := \frac{Q+Q^T}{2}$, by the fact that $x^TQx = x^T(\frac{Q+Q^T}{2})x$
- Compute eigenvalue decomposition:

$$Q = U\Gamma U^T$$

- Partition columns of U by $I^+ \in \mathbb{R}^{n\times n}, I^+_{ii} = 1$ if $\Gamma_i > 0$
- We define $U^+ = U \cdot \sqrt{(I^+)\cdot\Gamma},\ U^- = U \cdot \sqrt{-(I^-)\cdot\Gamma}$
- We have:

$$Q = U^+(U^+)^T - U^-(U^-)^T$$

1.2.2 Many-small-cone Relaxation

Now we conclude the many-small-cone relaxation,

$$\begin{aligned}
\text{Maximize} \quad & (y^+)^Te - (y^-)^Te + q^Tx \\
\text{s.t.} \quad & \begin{bmatrix} y_i^+ & z_i^+ \\ z_i^+ & 1 \end{bmatrix} \succeq 0, \begin{bmatrix} y_i^- & z_i^- \\ z_i^- & 1 \end{bmatrix} \succeq 0 \qquad && \forall i \\
& (U^+)^Tx = z^+, (U^-)^Tx = z^- \\
& \begin{bmatrix} Y_{j,i}^+ & Z_{j,i}^+ \\ Z_{j,i}^+ & 1 \end{bmatrix} \succeq 0, \begin{bmatrix} Y_{j,i}^- & Z_{j,i}^- \\ Z_{j,i}^- & 1 \end{bmatrix} \succeq 0 && \forall j, \forall i \\
& (U_j^+)^Tx = Z_j^+, (U_j^-)^Tx = Z_j^- && \forall j \\
& (Y_j^+)^Te - (Y_j^-)^Te + a_j^Tx(\le, =, \ge)\ b_j && \forall j
\end{aligned} \tag{1.8}$$

where $Q = U^+(U^+)^T - U^-(U^-)^T, A_j = U_j^+(U_j^+)^T - U_j^-(U_j^-)^T, j = 1, ..., m$

1.2.3 Bounds

To avoid unboundedness (the above is unbounded above), for example,

$$(U^+)^Tx = z^+, y^+ = (U^+)^T$$

1.3 Remark

1.3.1 Extending to matrix and tensor case

We first develop for the vector case: $x \in \mathbb{R}^n$, whereas QCQP is not limited to vector case:

- vectors, $x \in \mathbb{R}^n$, max-cut, quadratic knapsack problem
- matrices, $x \in \mathbb{R}^{n \times d}$, quadratic assignment problem, SNL, kissing number.

For example, SNL uses $X \in \mathbb{R}^{n \times d}$ for d-dimensional coordinates. For higher dimensional case, followings can be done:

- One may however using the vectorized method, i.e., $x = \texttt{vec}(X)$ to reformulate the matrix-based optimization problem, given the SDP bounds by original and vectorized relaxations are equivalent with mild assumptions. (see Ding et al. (2011))
- the above method may create a matrix of very large dimension resulted from Kronecker product.
- ultimately, the solver should provide an option to use user specified relaxations.

1.4 Tests

We test on specific applications:

- vectors, $x \in \mathbb{R}^n$, max-cut, quadratic knapsack problem
- matrices, $x \in \mathbb{R}^{n \times d}$, QAP, SNL, kissing number.

and a recent general new library as present in `qplib`, `http://qplib.zib.de/instances.html`, see Furini et al. (2019).

2 Branch and Cut Algorithm

For $x \in \mathbb{R}^n$, we have: $x^T A_i x = A_i \bullet (xx^T)$

$$
\begin{aligned}
\text{Maximize} \quad & Q \bullet Y \\
\text{s.t.} \quad & Y - xx^T \succeq 0 \text{ or } \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0 \\
& A_i \bullet Y \ (\leq, =, \geq)\ b_i, \forall i
\end{aligned} \tag{2.1}
$$

For MINLP and more specifically for QP, see Belotti et al. (2013), Misener and Floudas (2013). More recently, spacial branch-and-cut method Chen et al. (2017) for QP with complex variables.

Branching node and value selection, namely, select which $x_i, i = 1, ..., N$ and value α, for left and right children, for example:

$$x_i \leq \alpha, x_i \geq \alpha$$

- Audet et al. (2000) using RLT relaxation and LP, literally, $W_{ij} \approx x_i x_j, v_i \approx x_i^2$. The branching is essentially based on $\|W_{ij} - x_i x_j\|, \|v_i - x_i^2\|$, at each node we solve a linear programming relaxation.
- Linderoth (2005) a B-B based on subdividing the feasible region into the Cartesian product of triangles and rectangles.

Branching value:

Some source code to look at:

- Couenne: https://www.coin-or.org/Couenne/
-

2.1 Root

The root of the problem can selected from different SDPs as we discussed before. It is a question to answer whether we should use the SDP with the tightest bound.

References

Audet C, Hansen P, Jaumard B, Savard G (2000) A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming* 87(1):131–152, publisher: Springer.

Belotti P, Kirches C, Leyffer S, Linderoth J, Luedtke J, Mahajan A (2013) Mixed-integer nonlinear optimization. *Acta Numerica* 22:1, publisher: Cambridge University Press.

Chen C, Atamtürk A, Oren SS (2017) A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables. *Mathematical Programming* 165(2):549–577, publisher: Springer.

Ding Y, Ge D, Wolkowicz H (2011) On equivalence of semidefinite relaxations for quadratic matrix programming. *Mathematics of Operations Research* 36(1):88–104, publisher: INFORMS.

Furini F, Traversi E, Belotti P, Frangioni A, Gleixner A, Gould N, Liberti L, Lodi A, Misener R, Mittelmann H, Sahinidis NV, Vigerske S, Wiegele A (2019) QPLIB: a library of quadratic programming instances. *Mathematical Programming Computation* 11(2), ISSN 1867-2949, URL http://dx.doi.org/10.1007/s12532-018-0147-4.

Linderoth J (2005) A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical programming* 103(2):251–282, publisher: Springer.

Misener R, Floudas CA (2013) GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization* 57(1):3–50, publisher: Springer.

Appendix