

A benchmark generator for boolean quadratic programming

Michael X. Zhou

Received: date / Accepted: date

Abstract For boolean quadratic programming (BQP), we will show that there is no duality gap between the primal and dual problems under some conditions by using the classical Lagrangian duality. A benchmark generator is given to create random BQP problems which can be solved in polynomial time. Several numerical examples are generated to demonstrate the effectiveness of the proposed method.

Keywords Lagrangian duality · Boolean quadratic programming · Polynomial time · Benchmark generator

1 Theoretical basics

Considering the following boolean quadratic programming (BQP) problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} - \mathbf{c}^T \mathbf{x} \mid \mathbf{x} \in \{-1, 1\}^n \right\}, \quad (1)$$

where, $Q = Q^T \in \mathbb{R}^{n \times n}$ is a given indefinite matrix, $\mathbf{c} \in \mathbb{R}^n$ is a given nonzero vector.

By introducing the Lagrange multiplier λ_i associated with each constraint $\frac{1}{2}(x_i^2 - 1) = 0$, the Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ can be defined as

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \frac{1}{2} \mathbf{x}^T Q \mathbf{x} - \mathbf{c}^T \mathbf{x} + \sum_{i=1}^n \frac{1}{2} \lambda_i (x_i^2 - 1) \\ &= \frac{1}{2} \mathbf{x}^T Q(\boldsymbol{\lambda}) \mathbf{x} - \mathbf{c}^T \mathbf{x} - \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{e} \end{aligned} \quad (2)$$

Michael X. Zhou
School of Information Science and Engineering, Central South University, Changsha 410083, China.

where, $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)^T$, $\mathbf{e} \in \mathbb{R}^n$ is a vector with all entries one, and $Q_{\boldsymbol{\lambda}}$ is

$$Q(\boldsymbol{\lambda}) = Q + \text{diag}(\boldsymbol{\lambda}), \quad (3)$$

here, $\text{diag}(\boldsymbol{\lambda})$ is a diagonal matrix with its diagonal entries $\lambda_1, \dots, \lambda_n$.

The Lagrangian dual function can be obtained by

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{x} \in \{-1, 1\}^n} L(\mathbf{x}, \boldsymbol{\lambda}). \quad (4)$$

Let define the following dual feasible space

$$\mathcal{S}^+ = \{\boldsymbol{\lambda} \in \mathbb{R}^n | Q(\boldsymbol{\lambda}) \succ 0\}, \quad (5)$$

then the Lagrangian dual function can be written explicitly as

$$g(\boldsymbol{\lambda}) = -\frac{1}{2} \mathbf{c}^T Q^{-1}(\boldsymbol{\lambda}) \mathbf{c} - \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{e}, \quad \text{s.t. } \boldsymbol{\lambda} \in \mathcal{S}^+ \quad (6)$$

associated with the Lagrangian equation

$$Q(\boldsymbol{\lambda}) \mathbf{x} = \mathbf{c}. \quad (7)$$

Finally, the Lagrangian dual problem can be obtained as

$$\max_{\boldsymbol{\lambda} \in \mathcal{S}^+} \left\{ g(\boldsymbol{\lambda}) = -\frac{1}{2} \mathbf{c}^T Q^{-1}(\boldsymbol{\lambda}) \mathbf{c} - \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{e} \right\}. \quad (8)$$

Theorem 1 *If $\bar{\boldsymbol{\lambda}}$ is a critical point of the Lagrangian dual function and $\bar{\boldsymbol{\lambda}} \in \mathcal{S}^+$, then the corresponding $\bar{\mathbf{x}} = Q^{-1}(\bar{\boldsymbol{\lambda}}) \mathbf{c}$ is a global solution to the BQP problem.*

Proof The derivative of $g(\boldsymbol{\lambda})$ gives

$$\begin{aligned} \frac{\partial g(\boldsymbol{\lambda})}{\partial \bar{\lambda}_i} &= -\frac{1}{2} \frac{\partial [\mathbf{c}^T Q^{-1}(\boldsymbol{\lambda}) \mathbf{c}]}{\partial \bar{\lambda}_i} - \frac{1}{2} \\ &= \frac{1}{2} \mathbf{c}^T Q^{-1}(\bar{\boldsymbol{\lambda}}) \frac{\partial Q(\boldsymbol{\lambda})}{\partial \bar{\lambda}_i} Q^{-1}(\bar{\boldsymbol{\lambda}}) \mathbf{c} - \frac{1}{2} \\ &= \frac{1}{2} \bar{x}_i^2 - \frac{1}{2}, \end{aligned}$$

where, $\bar{\mathbf{x}} = Q^{-1}(\bar{\boldsymbol{\lambda}}) \mathbf{c}$. Since $\bar{\boldsymbol{\lambda}}$ is a critical point of the Lagrangian dual function, we have $\frac{1}{2} \bar{x}_i^2 - \frac{1}{2} = 0$.

To continue, we have

$$\begin{aligned} f(\mathbf{x}) - f(\bar{\mathbf{x}}) &= \frac{1}{2} \mathbf{x}^T Q \mathbf{x} - \mathbf{c}^T \mathbf{x} - \frac{1}{2} \bar{\mathbf{x}}^T Q \bar{\mathbf{x}} + \mathbf{c}^T \bar{\mathbf{x}} - \sum_{i=1}^n \frac{1}{2} \bar{\lambda}_i (\bar{x}_i^2 - 1) \\ &= \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T Q(\bar{\boldsymbol{\lambda}}) (\mathbf{x} - \bar{\mathbf{x}}) - \frac{1}{2} \mathbf{x}^T \text{diag}(\bar{\boldsymbol{\lambda}}) \mathbf{x} + \mathbf{x}^T Q(\bar{\boldsymbol{\lambda}}) \bar{\mathbf{x}} - \bar{\mathbf{x}}^T Q(\bar{\boldsymbol{\lambda}}) \bar{\mathbf{x}} \\ &\quad - \mathbf{c}^T (\mathbf{x} - \bar{\mathbf{x}}) + \frac{1}{2} \bar{\boldsymbol{\lambda}}^T \mathbf{e} \\ &= \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T Q(\bar{\boldsymbol{\lambda}}) (\mathbf{x} - \bar{\mathbf{x}}) + (\mathbf{x} - \bar{\mathbf{x}})^T (Q(\bar{\boldsymbol{\lambda}}) \bar{\mathbf{x}} - \mathbf{c}) \\ &= \frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T Q(\bar{\boldsymbol{\lambda}}) (\mathbf{x} - \bar{\mathbf{x}}) > 0, \quad \forall \mathbf{x} \in \{-1, 1\}^n \end{aligned}$$

that is to say, $\bar{\mathbf{x}} = Q^{-1}(\bar{\boldsymbol{\lambda}})\mathbf{c}$ is the global minimizer of the BQP problem. This completes the proof. \square

Remark 1 Under the conditions in Theorem 1, it is easy to verify that

$$\begin{aligned}
g(\bar{\boldsymbol{\lambda}}) &= -\frac{1}{2}\mathbf{c}^T Q^{-1}(\bar{\boldsymbol{\lambda}})\mathbf{c} - \frac{1}{2}\bar{\boldsymbol{\lambda}}^T \mathbf{e} \\
&= -\frac{1}{2}\bar{\mathbf{x}}^T Q(\bar{\boldsymbol{\lambda}})\bar{\mathbf{x}} + (\bar{\mathbf{x}}^T Q(\bar{\boldsymbol{\lambda}})\bar{\mathbf{x}} - \mathbf{c}^T \bar{\mathbf{x}}) - \frac{1}{2}\bar{\boldsymbol{\lambda}}^T \mathbf{e} \\
&= \frac{1}{2}\bar{\mathbf{x}}^T Q(\bar{\boldsymbol{\lambda}})\bar{\mathbf{x}} - \mathbf{c}^T \bar{\mathbf{x}} - \frac{1}{2}\bar{\boldsymbol{\lambda}}^T \mathbf{e} = L(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}) \\
&= \frac{1}{2}\bar{\mathbf{x}}^T Q\bar{\mathbf{x}} - \mathbf{c}^T \bar{\mathbf{x}} + \sum_{i=1}^n \frac{1}{2}\bar{\lambda}_i(\bar{x}_i^2 - 1) \\
&= f(\bar{\mathbf{x}}),
\end{aligned}$$

which shows that there is no duality gap between the primal and dual problems.

2 A benchmark generator for BQP

As stated above, if there exists a critical point of Lagrangian dual function in \mathcal{S}^+ , zero duality gap will be met. In this section, we will construct a benchmark generator for BQP problem such that these conditions are satisfied.

The inverse problem can be simplified as follows

$$\begin{aligned}
&\text{find} \quad Q, \mathbf{c}, \mathbf{x}, \boldsymbol{\lambda} \\
&\text{s.t.} \quad (Q + \text{diag}(\boldsymbol{\lambda}))\mathbf{x} = \mathbf{c} \\
&\quad \quad Q + \text{diag}(\boldsymbol{\lambda}) \succ 0 \\
&\quad \quad \mathbf{x} \in \{-1, 1\}^n
\end{aligned} \tag{9}$$

where, $Q = Q^T \in \mathbb{R}^{n \times n}$, $\mathbf{c}, \mathbf{x}, \boldsymbol{\lambda} \in \mathbb{R}^n$.

Suppose Q is a freely random symmetric matrix, to guarantee $Q + \text{diag}(\boldsymbol{\lambda}) \succ 0$, let $\boldsymbol{\lambda}$ satisfy

$$\lambda_i \geq \sum_{j=1}^n |Q_{ij}|, \tag{10}$$

to make sure that $Q(\boldsymbol{\lambda})$ is a diagonally dominant matrix.

Suppose \mathbf{x} is a freely random “true” solution, then \mathbf{c} should satisfy

$$\mathbf{c} = (Q + \text{diag}(\boldsymbol{\lambda}))\mathbf{x}. \tag{11}$$

The matlab scripts for generating a benchmark BQP are given in the following

```

function [Q,c,x,lambda] = generate_Qc(n)
base = 10;
Q = base*randn(n);
Q = round((Q + Q')/2);
lambda = zeros(n,1);
x = round(rand(n,1));
x = 2*x - 1;
lambda = sum(abs(Q),2);
c = (Q + diag(lambda))*x;

```

where, *base* is set to control the range of elements in Q .

3 Numerical experiments

The Lagrangian dual problem can be reformulated as the following SDP problem easily via Schur complement [1]

$$\begin{aligned}
 (\text{SDP}) : \min \quad & \frac{1}{2}t + \frac{1}{2}\boldsymbol{\lambda}^T \mathbf{e} \\
 \text{s.t.} \quad & \begin{pmatrix} Q(\boldsymbol{\lambda}) & \mathbf{c} \\ \mathbf{c}^T & t \end{pmatrix} \succeq 0
 \end{aligned} \tag{12}$$

In the next, several examples are created by the above mentioned generator function *generate_Qc*(\cdot). All of the experiments are run on MATLAB R2010b on Intel(R) Core(TM) i3-2310M CPU @2.10GHz under Window 7 environment. The SDPT3 [3] is used as a solver embedded in YALMIP [2] for the SDP problem.

Example 1

$$Q = \begin{pmatrix} -4 & -3 & 6 & -3 & -6 \\ -3 & 13 & -25 & -5 & 3 \\ 6 & -25 & -2 & -5 & -1 \\ -3 & -5 & -5 & -8 & -7 \\ -6 & 3 & -1 & -7 & -5 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} -18 \\ 92 \\ -62 \\ -10 \\ 0 \end{pmatrix}$$

By solving the corresponding Lagrangian dual problem in 0.9204 seconds, we can get $\bar{\boldsymbol{\lambda}} = (21.9996, 48.9999, 39.0000, 27.9996, 21.9998)^T$ and $\bar{\mathbf{x}} = Q^{-1}(\bar{\boldsymbol{\lambda}})\mathbf{c} = (-1.0000, 1.0000, -1.0000, -1.0000, -1.0000)^T$.

Example 2

$$Q = \begin{pmatrix} -6 & -9 & -7 & -3 & 2 & -2 & -12 & -11 & 8 & -6 \\ -9 & 14 & -4 & 3 & -2 & -4 & 6 & 23 & 8 & 3 \\ -7 & -4 & 1 & 21 & -10 & 2 & 6 & -13 & -9 & 4 \\ -3 & 3 & 21 & -18 & -2 & 7 & -2 & 16 & 1 & 3 \\ 2 & -2 & -10 & -2 & 10 & 8 & -11 & -3 & -2 & -2 \\ -2 & -4 & 2 & 7 & 8 & 0 & -10 & -10 & -2 & -7 \\ -12 & 6 & 6 & -2 & -11 & -10 & -7 & -10 & 3 & 0 \\ -11 & 23 & -13 & 16 & -3 & -10 & -10 & 6 & -8 & 10 \\ 8 & 8 & -9 & 1 & -2 & -2 & 3 & -8 & -10 & -5 \\ -6 & 3 & 4 & 3 & -2 & -7 & 0 & 10 & -5 & -9 \end{pmatrix}, \quad c = \begin{pmatrix} 24 \\ -84 \\ 72 \\ -18 \\ -72 \\ 16 \\ 38 \\ 54 \\ -66 \\ 42 \end{pmatrix}$$

By solving the corresponding Lagrangian dual problem in 0.9204 seconds, we can get $\bar{\lambda} = (66.0010, 75.9997, 76.9970, 76.0010, 51.9993, 51.9988, 66.9988, 109.9979, 55.9985, 48.9995)^T$ and $\bar{x} = Q^{-1}(\bar{\lambda})c = (1.0000, -1.0000, 1.0000, -1.0000, -1.0000, 1.0000, 1.0000, 1.0000, 1.0000, -1.0000)^T$.

Example 3

$$Q = \begin{pmatrix} 11 & 2 & -10 & 15 & 21 & -7 & -8 & 6 & -3 & 11 & 2 & -1 & 3 & -2 & -1 \\ 2 & 6 & 7 & -5 & 10 & -14 & -1 & -8 & 3 & 6 & 6 & 0 & -7 & 1 & -2 \\ -10 & 7 & 21 & 12 & 13 & -9 & -1 & 2 & -5 & 9 & 2 & -1 & -2 & 4 & 8 \\ 15 & -5 & 12 & 12 & -7 & 0 & -3 & -17 & -3 & 6 & -1 & -1 & -1 & 6 & -5 \\ 21 & 10 & 13 & -7 & 3 & 6 & 3 & -1 & -10 & 0 & -9 & -1 & -4 & -7 & -2 \\ -7 & -14 & -9 & 0 & 6 & 5 & 1 & 7 & 3 & 2 & -1 & 3 & -4 & 3 & 8 \\ -8 & -1 & -1 & -3 & 3 & 1 & -5 & -5 & 3 & 6 & 17 & -13 & 6 & 14 & -10 \\ 6 & -8 & 2 & -17 & -1 & 7 & -5 & -2 & -6 & 1 & 12 & 0 & 5 & 5 & -4 \\ -3 & 3 & -5 & -3 & -10 & 3 & 3 & -6 & -26 & 3 & -4 & 7 & 13 & -4 & -2 \\ 11 & 6 & 9 & 6 & 0 & 2 & 6 & 1 & 3 & 13 & -11 & 10 & -12 & -13 & -11 \\ 2 & 6 & 2 & -1 & -9 & -1 & 17 & 12 & -4 & -11 & -1 & 12 & -9 & 7 & 5 \\ -1 & 0 & -1 & -1 & -1 & 3 & -13 & 0 & 7 & 10 & 12 & -1 & -1 & 5 & 9 \\ 3 & -7 & -2 & -1 & -4 & -4 & 6 & 5 & 13 & -12 & -9 & -1 & -15 & 6 & -1 \\ -2 & 1 & 4 & 6 & -7 & 3 & 14 & 5 & -4 & -13 & 7 & 5 & 6 & 15 & 10 \\ -1 & -2 & 8 & -5 & -2 & 8 & -10 & -4 & -2 & -11 & 5 & 9 & -1 & 10 & 8 \end{pmatrix},$$

$$c = (-140, 86, -118, -114, -134, -72, 92, -100, 120, 98, -80, 70, 90, 120, 78)^T.$$

By solving the corresponding Lagrangian dual problem in 0.9204 seconds, we can get $\bar{\lambda} = (102.9966, 77.9974, 105.9976, 93.9972, 96.9967, 72.9976, 95.9974, 80.9970, 94.9967, 113.9981, 98.9980, 64.9976, 88.9972, 101.9973, 85.9979)^T$ and $\bar{x} = Q^{-1}(\bar{\lambda})c = (-1.0000, 1.0000, -1.0000, -1.0000, -1.0000, -1.0000, 1.0000, 1.0000, -1.0000, 1.0000, -1.0000, 1.0000, 1.0000, 1.0000, 1.0000)^T$.

Remark 2 All of the tested problems are solved in 0.9204 seconds, which can be regarded as an indicator for polynomial time complexity.

Example 4 Other large random BQP problems are created by the same procedure, and the corresponding running time for solving these problems can be found in Table 1.

Table 1 Running time for other large random BQP cases

size	100	200	300	400	500	600	700	800	900	1000
time(s)	2.1840	2.6832	4.1652	6.8952	11.2321	15.9901	23.0101	33.1658	40.2015	56.4724

References

1. Stephen P. Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan, Linear matrix inequalities in system and control theory, Philadelphia: SIAM, 1994.
2. Johan Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, IEEE International Symposium on Computer Aided Control Systems Design, pp. 284–289, 2004.
3. Kim-Chuan Toh, Michael J. Todd and Reha H. Tütüncü, SDPT3-a MATLAB software package for semidefinite programming, version 1.3, Optimization methods and software, 11: 545–581, 1999.