Solving the Quadratic Eigenvalue Complementarity Problem by DC Programming

Yi-Shuai Niu[1], Joaquim Júdice[2], Le Thi Hoai An[3] and Pham Dinh Tao[4]

[1]Shanghai JiaoTong University, Maths Departement and SJTU-Paristech, China
niuyishuai@sjtu.edu.cn
[2]Instituto de Telecomunicações, Portugal
judice@co.it.pt
[3]University of Lorraine, France
hoai-an.le-thi@univ-lorraine.fr
[4]National Institute of Applied Sciences, France
pham@insa-rouen.fr

Abstract. We present in this paper some results for solving the Quadratic Eigenvalue Complementarity Problem (QEiCP) by using DC(Difference of Convex functions) programming approaches. Two equivalent Nonconvex Polynomial Programming (NLP) formulations of QEiCP are introduced. We focus on the construction of the DC programming formulations of the QEiCP from these NLPs. The corresponding numerical solution algorithms based on the classical DC Algorithm (DCA) are also discussed.

Keywords: Eigenvalue Problem, Complementarity Problem, Nonconvex Polynomial Programming, DC Programming, DCA.

1 Introduction

Given three matrices $A, B, C \in \mathbb{R}^{n\times n}$, the *Quadratic Eigenvalue Complementarity Problem (QEiCP)* consists of finding a $\lambda \in \mathbb{R}$ and an associated nonzero vector $x \in \mathbb{R}^n$ such that

$$\begin{gathered} w = \lambda^2 Ax + \lambda Bx + Cx \\ x^T w = 0 \\ x \geq 0, w \geq 0 \end{gathered} \tag{1}$$

This problem and some applications have been firstly introduced in [19] and is usually denoted by QEiCP(A, B, C). In any solution (λ, x) of QEiCP(A, B, C), the λ-component is called a *quadratic complementary eigenvalue*, and the vector x-component is a *quadratic complementary eigenvector* associated to λ.

QEiCP is an extension of the well-known Eigenvalue Complementarity Problem (EiCP) [18], which consists of finding a complementary eigenvalue $\lambda \in \mathbb{R}$

and an associated complementary eigenvector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$\begin{array}{c} w = \lambda B x - C x \\ x^T w = 0 \\ x \geq 0, w \geq 0 \end{array} \tag{2}$$

where $B, C \in \mathbb{R}^{n\times n}$ are two given matrices.

Clearly, EiCP is a special case of QEiCP where the matrix A is null. During the past several years, many applications of EiCP have been discussed and a number of algorithms have been proposed for the solution of this problem and some extensions [1–3, 6–11, 15, 16].

EiCP has at least one solution if the matrix B of the leading λ-term is positive definite (PD) [9, 18]. Contrary to the EiCP, the QEiCP may have no solution even when the matrix A of leading λ-term is PD. For instance, if $B = 0$, A, C are PD matrices, there is no solution for QEiCP since $x^T w = \lambda^2 x^T Ax + x^T Cx > 0, \forall \lambda \in \mathbb{R}, x \in \mathbb{R}^n \setminus \{0\}$.

The existence of a solution to QEiCP depends on the given (A, B, C). If the matrix A is PD, QEiCP has at least a solution if one of the two following conditions holds:

(i) $C \notin S_0$ [4], where S_0 is the class of matrices defined by

$$C \in S_0 \Leftrightarrow \exists x \geq 0, x \neq 0, Cx \geq 0.$$

(ii) *co-hyperbolicity*[19] : $(x^T Bx)^2 \geq 4(x^T Ax)(x^T Cx)$ for all $x \geq 0, x \neq 0$.

In practice, investigating whether $C \in S_0$ reduces to solving a special linear program [4]. On the other hand, it is relatively hard to prove that co-hyperbolicity holds. However, there are some sufficient conditions which imply the co-hyperbolicity. For instance, this occurs if A and $-C$ are both PD matrices.

A number of algorithms have been proposed for the solution of QEiCP when $A \in$PD and one of the conditions $C \notin S_0$ or co-hyperbolicity holds [1, 4–6, 19]. As discussed in [4–6], some of these methods are based on nonlinear programming (NLP) formulations of QEiCP such that (λ, x) is a solution of QEiCP if and only if (λ, x) is a global minimum of NLP with an optimal value equal to zero. In this paper, we introduce two nonlinear programming formulations and their corresponding DC programming formulations when co-hyperbolicity holds, and we briefly discuss the DC Algorithm for the solution of these DC programs.

The paper is organized as follows. Section 2 contains the nonlinear programming formulations of QEiCP, and the corresponding dc formulations mentioned before. A new result on lower and upper bounds estimation of the quadratic complementary eigenvalue is given in section 3. The numerical solution algorithms for solving these DC programming formulations are discussed in section 4. Some conclusions are presented in the last section.

2 DC Programming Formulations for QEiCP

In this section, we introduce two DC programming formulations of QEiCP when $A \in$PD and the co-hyperbolic property holds. These DC programs are based on two NLP formulations of QEiCP. The construction of the DC programming problem requires lower and upper bounds on the λ-variable which can be computed by the procedures discussed in [6]. We will also present a new procedure for such a goal in the section 3.

2.1 Nonlinear Programming Formulations

As discussed in [6], QEiCP is equivalent to the following NLP:

$$(P) \qquad \begin{aligned} 0 = \min\ & f(x,y,z,w,\lambda) := \|y-\lambda x\|^2 + \|z-\lambda y\|^2 + x^T w \\ s.t.\ & w = Az + By + Cx \\ & e^T x = 1 \\ & e^T y = \lambda \\ & x \geq 0, w \geq 0, z \geq 0. \end{aligned} \tag{3}$$

As (x, y, z, w, λ) is an optimal solution of the problem (P) if and only if (λ, x) is a solution of QEiCP. In fact, for any solution of QEiCP (λ, x) that does not satisfy $e^T x = 1$, we can always construct a solution $(\lambda, \frac{x}{e^T x})$ of QEiCP satisfying such a constraint.

The problem (P) is a polynomial programming problem where a nonconvex polynomial function $f(x, y, z, w, \lambda)$ is minimized subject to linear constraints. Due to the fact that any polynomial function is a dc function, we can reformulate the problem (P) as a dc program.

On the other hand, observing that the complementarity constraint $w^T x = 0, x \geq 0, w \geq 0$ holds if and only if $w^T x = \sum_{i=1}^n \min(x_i, w_i) = 0$, we have the following equivalent nonlinear programming formulation of (P):

$$(P') \quad \begin{aligned} 0 = \min\ & f'(x,y,z,w,\lambda) = \|y-\lambda x\|^2 + \|z-\lambda y\|^2 + \textstyle\sum_{i=1}^n \min(x_i, w_i) \\ s.t\ & w = Az + By + Cx \\ & e^T x = 1 \\ & e^T y = \lambda \\ & x \geq 0, w \geq 0, z \geq 0. \end{aligned}$$

The problems (P) and (P') have the same set of linear constraints. The difficulty for solving (P) and (P') relies on the non-convexity on their objective functions.

2.2 DC programming formulations

The polynomial function f in (P) can be decomposed into four parts:

$$\begin{aligned} f(x,y,z,w,\lambda) &= \|y\|^2 + \|z\|^2 - 2\lambda y^T(x+z) + \lambda^2(\|x\|^2 + \|y\|^2) + x^T w \\ &= f_0(y,z) + f_1(x,y,z,\lambda) + f_2(x,y,\lambda) + f_3(x,w) \end{aligned}$$

with

$$\begin{cases} f_0(y,z) = \|y\|^2 + \|z\|^2 \\ f_1(x,y,z,\lambda) = -2\lambda y^T(x+z) \\ f_2(x,y,\lambda) = \lambda^2(\|x\|^2 + \|y\|^2) \\ f_3(x,w) = x^T w \end{cases}$$

The function f_0 is convex quadratic function, while f_1, f_2, f_3 are nonconvex polynomial functions. Similarly, the objective function f' in (P') is also decomposed into the following four terms as:

$$f'(x,y,z,w,\lambda) = f_0(y,z) + f_1(x,y,z,\lambda) + f_2(x,y,\lambda) + \tilde{f}_3(x,w)$$

where $\tilde{f}_3(x,w)$ defined by $\sum_{i=1}^n \min(x_i,w_i)$ is a polyhedral concave function.

Both the bilinear function f_3 and the polyhedral concave function $\tilde{f}_3$ are classical dc functions whose dc decompositions are as follows:

1. DC decomposition of bilinear function f_3:
$$f_3(x,w) = \frac{\|x+w\|^2}{4} - \frac{\|x-w\|^2}{4} \tag{4}$$
in which $\frac{\|x+w\|^2}{4}$ and $\frac{\|x-w\|^2}{4}$ are both convex quadratic functions.
2. DC decomposition of polyhedral function $\tilde{f}_3$:
$$\tilde{f}_3(x,w) = \sum_{i=1}^n \min(x_i,w_i) = (0) - (-\sum_{i=1}^n \min(x_i,w_i)) \tag{5}$$
where $-\sum_{i=1}^n \min(x_i,w_i)$ is a convex polyhedral function.

To obtain a dc decompositions of the nonconvex polynomial functions f_1 and f_2, we first obtain the expressions of their gradients and hessians:

1. Gradient and Hessian of f_1:
$$\nabla f_1(x,y,z,\lambda) = \begin{bmatrix} \nabla_x f_1(x,y,z,\lambda) \\ \nabla_y f_1(x,y,z,\lambda) \\ \nabla_z f_1(x,y,z,\lambda) \\ \nabla_\lambda f_1(x,y,z,\lambda) \end{bmatrix} = \begin{bmatrix} -2\lambda y \\ -2\lambda(x+z) \\ -2\lambda y \\ -2y^T(x+z) \end{bmatrix}.$$
$$\nabla^2 f_1(x,y,z,\lambda) = \begin{bmatrix} 0 & -2\lambda I & 0 & -2y \\ -2\lambda I & 0 & -2\lambda I & -2(x+z) \\ 0 & -2\lambda I & 0 & -2y \\ -2y^T & -2(x+z)^T & -2y^T & 0 \end{bmatrix}.$$
2. Gradient and Hessian of f_2:
$$\nabla f_2(x,y,\lambda) = \begin{bmatrix} \nabla_x f_2(x,y,\lambda) \\ \nabla_y f_2(x,y,\lambda) \\ \nabla_\lambda f_2(x,y,\lambda) \end{bmatrix} = \begin{bmatrix} 2\lambda^2 x \\ 2\lambda^2 y \\ 2\lambda(\|x\|^2 + \|y\|^2) \end{bmatrix}.$$
$$\nabla^2 f_2(x,y,z,\lambda) = \begin{bmatrix} 2\lambda^2 I & 0 & 4\lambda x \\ 0 & 2\lambda^2 I & 4\lambda y \\ 4\lambda x^T & 4\lambda y^T & 2(\|x\|^2 + \|y\|^2) \end{bmatrix}$$

The spectral radius of the hessian matrices $\nabla^2 f_1$ and $\nabla^2 f_2$ (denoted by $\rho(\nabla^2 f_1)$ and $\rho(\nabla^2 f_2)$) can be bounded above by the induced 1-norm as follows:

$$\rho(\nabla^2 f_1) \leq \|\nabla^2 f_1\|_1 = 2\max\{|\lambda| + |y_i|, |x_i + z_i| + 2|\lambda|, \sum_i (2|y_i| + |x_i + z_i|)\}$$

$$\rho(\nabla^2 f_2) \leq \|\nabla^2 f_2\|_1 = 2\max\{\lambda^2 + 2|\lambda||x_i|, \lambda^2 + 2|\lambda||y_i|, \|x\|^2 + \|y\|^2 + 2|\lambda| \sum_i (|x_i| + |y_i|)\}$$

Thus $\rho(\nabla^2 f_1)$ and $\rho(\nabla^2 f_2)$) are bounded when the variables (x, y, z, w, λ) of (P) and (P') are bounded.

The next proposition shows that if the quadratic complementary eigenvalue λ of QEiCP is bounded, then the variables x, y, z, w are bounded with respect to the bounds of λ.

Proposition 1. *If the quadratic complementary eigenvalue λ of QEiCP is bounded in an interval $[l, u]$, then any optimal solution of (P) and (P') satisfies:*

$$x \in [0,1]^n; y \in [\min\{0, l\}, \max\{0, u\}]^n; z \in [0, \max\{u^2, l^2\}]^n;$$

$$0 \leq w \leq \begin{bmatrix} \max\{u^2, l^2\} \sum_j |A_{1j}| + \max\{|l|, |u|\} \sum_j |B_{1j}| + \sum_j |C_{1j}| \\ \vdots \\ \max\{u^2, l^2\} \sum_j |A_{nj}| + \max\{|l|, |u|\} \sum_j |B_{nj}| + \sum_j |C_{nj}| \end{bmatrix}.$$

Proof. Suppose that we could determine some values l and u such that λ-component of QEiCP is located in the interval $[l, u]$.

1. $e^T x = 1, x \geq 0$ implies $x \in [0,1]^n$.
2. $y = \lambda x, x \in [0,1]^n$ and $\lambda \in [l, u]$ imply $y \in [\min\{0, l\}, \max\{0, u\}]^n$.
3. $z = \lambda y, y = \lambda x \Rightarrow z = \lambda^2 x$, with $x \in [0,1]^n, \lambda \in [l, u]$, leads to $z \in [0, \max\{u^2, l^2\}]^n$.
4. Since $w \geq 0$, the upper bound of w is obtained from the definition of w as $Az+By+Cx$. As $x \in [0,1]^n, y \in [\min\{0, l\}, \max\{0, u\}]^n, z \in [0, \max\{u^2, l^2\}]^n$, then w is also bounded:

$$|w| \leq \begin{bmatrix} \max\{u^2, l^2\} \sum_j |A_{1j}| + \max\{|l|, |u|\} \sum_j |B_{1j}| + \sum_j |C_{1j}| \\ \vdots \\ \max\{u^2, l^2\} \sum_j |A_{nj}| + \max\{|l|, |u|\} \sum_j |B_{nj}| + \sum_j |C_{nj}| \end{bmatrix}.$$

□

Let us define the convex polyhedral set:

$$\mathcal{C} := \{(x, y, z, w, \lambda) : w = Az + By + Cx, e^T x = 1, e^T y = \lambda, x \in [0,1]^n, \\ y \in [\min\{0, l\}, \max\{0, u\}]^n, z \in [0, \max\{u^2, l^2\}]^n, w \geq 0, l \leq \lambda \leq u\}.$$

The problems (P) and (P') defined on $\mathcal{C}$ have the same set of optimal solutions, and $\rho(\nabla^2 f_1)$ and $\rho(\nabla^2 f_2)$ are bounded. In fact, the following proposition holds:

Proposition 2. *For* $(x, y, z, w, \lambda) \in \mathcal{C}$,

$$\rho(\nabla^2 f_1) \le 2 + 2n(p^2 + 2p) = \rho_1$$

$$\rho(\nabla^2 f_2) \le 2(3np^2 + 2p + 1) = \rho_2$$

where $p = \max\{|l|, |u|\}$.

Proof. Since $\lambda \in [l, u]$, then $|\lambda| \le \max\{|l|, |u|\} = p$. Hence,

$$\rho(\nabla^2 f_1) \le 2\max\{|\lambda| + |y_i|, |x_i + z_i| + 2|\lambda|, \sum_i (2|y_i| + |x_i + z_i|)\}.$$

But,

$$\sum_i |y_i| \le np.$$

$$\sum_i |x_i + z_i| \le \sum_i |x_i| + \sum_i |z_i| \le 1 + np^2.$$

Hence,

$$\rho(\nabla^2 f_1) \le 2\max\{2p, 1 + p^2 + 2p, 1 + n(p^2 + 2p)\} = 2 + 2n(p^2 + 2p) = \rho_1$$

Similarly,

$$\rho(\nabla^2 f_2) \le 2\max\{\lambda^2 + 2|\lambda||x_i|, \lambda^2 + 2|\lambda||y_i|, \|x\|^2 + \|y\|^2 + 2|\lambda| \sum_i (|x_i| + |y_i|)\}$$

$$\le 2\max\{p^2 + 2p, 3p^2, 3np^2 + 2p + 1\} = 2(3np^2 + 2p + 1) = \rho_2.$$

□

Thus, we get a dc decomposition for f_1 and f_2 as follows:

$$f_1(x, y, z, \lambda) = \frac{\rho_1}{2}\|(x, y, z, \lambda)\|^2 - (\frac{\rho_1}{2}\|(x, y, z, \lambda)\|^2 - f_1(x, y, z, \lambda))$$

$$f_2(x, y, \lambda) = \frac{\rho_2}{2}\|(x, y, \lambda)\|^2 - (\frac{\rho_2}{2}\|(x, y, \lambda)\|^2 - f_2(x, y, \lambda))$$

where $\frac{\rho_1}{2}\|(x, y, z, \lambda)\|^2$ and $\frac{\rho_2}{2}\|(x, y, \lambda)\|^2$ are quadratic convex functions. While $\frac{\rho_1}{2}\|(x, y, z, \lambda)\|^2 - f_1(x, y, z, \lambda)$ and $\frac{\rho_2}{2}\|(x, y, \lambda)\|^2 - f_2(x, y, \lambda)$ are locally convex restricted on $\mathcal{C}$.

Using the dc decompositions of f_1,f_2,f_3 and $\tilde{f}_3$ derived in this section, we get the following dc decomposition for the objective functions f and f'.

1. A dc decomposition for f is given by:

$$f(x, y, z, w, \lambda) = g(x, y, z, w, \lambda) - h(x, y, z, w, \lambda)$$

   where

$$g(x, y, z, w, \lambda) = \frac{\|x + w\|^2}{4} + \frac{\rho_1 + \rho_2}{2}\|x\|^2 + (\frac{\rho_1 + \rho_2}{2} + 1)\|y\|^2 + (\frac{\rho_1}{2} + 1)\|z\|^2 + \frac{\rho_1 + \rho_2}{2}\lambda^2,$$

$$h(x, y, z, w, \lambda) = g(x, y, z, w, \lambda) - f(x, y, z, w, \lambda).$$

2. A dc decomposition for f' is given by:

$$f'(x,y,z,w,\lambda) = g'(x,y,z,w,\lambda) - h'(x,y,z,w,\lambda)$$

where

$$g'(x,y,z,\lambda) = \frac{\rho_1+\rho_2}{2}\|x\|^2 + (\frac{\rho_1+\rho_2}{2}+1)\|y\|^2 + (\frac{\rho_1}{2}+1)\|z\|^2 + \frac{\rho_1+\rho_2}{2}\lambda^2,$$

$$h'(x,y,z,w,\lambda) = g'(x,y,z,\lambda) - f'(x,y,z,w,\lambda).$$

The functions g and g' are both convex quadratic functions, while h and h' are locally convex functions restricted on the convex polyhedral set $\mathcal{C}$.

Finally, we get the following equivalent DC programs of (P) and (P') as below:

$$(P_{DC}) \qquad \begin{array}{c} 0 = \min\, g(x,y,z,w,\lambda) - h(x,y,z,w,\lambda) \\ s.t.\ (x,y,z,w,\lambda) \in \mathcal{C}. \end{array} \tag{6}$$

$$(P'_{DC}) \qquad \begin{array}{c} 0 = \min\, g'(x,y,z,\lambda) - h'(x,y,z,w,\lambda) \\ s.t.\ (x,y,z,w,\lambda) \in \mathcal{C}. \end{array} \tag{7}$$

3 Lower and upper bounds for the quadratic complementary eigenvalue $\boldsymbol{\lambda}$

Since the bounds of the variables x, y, z, w in $\mathcal{C}$, as well as the dc decompositions given in the previous section depend on the bounds of λ, we need to estimate its upper and lower bounds. The following theorem gives these values.

Proposition 3. *If $A \in$PD and the co-hyperbolic condition holds, the λ-component of any solution of QEiCP satisfies*

$$l = \beta - \sqrt{\alpha} \leq \lambda \leq \gamma + \sqrt{\alpha} = u$$

with $s = \min\{x^T A x : e^T x = 1, x \geq 0\}$, $\alpha = \max\{\gamma^2, \beta^2\} + \frac{\max_{i,j}\{-C_{ij}\}}{s}$,

$$\beta = \begin{cases} \frac{\min\{-B_{ij}\}}{2\max\{A_{ij}\}}, & \textit{if } \min\{-B_{ij}\} > 0; \\ \frac{\min\{-B_{ij}\}}{2s}, & \textit{if } \min\{-B_{ij}\} \leq 0. \end{cases}$$

$$\gamma = \begin{cases} \frac{\max\{-B_{ij}\}}{2s}, & \textit{if } \max\{-B_{ij}\} > 0; \\ \frac{\max\{-B_{ij}\}}{2\max\{A_{ij}\}}, & \textit{if } \max\{-B_{ij}\} \leq 0. \end{cases}$$

Proof. Since $A \in$PD and the co-hyperbolic condition holds, the λ-component of any solution of QEiCP satisfies

$$\lambda = \frac{-x^T B x \pm \sqrt{(x^T B x)^2 - 4(x^T A x)(x^T C x)}}{2 x^T A x}.$$

Let $U = \{e^T x = 1, x \geq 0\}$. For a given matrix $M \in \mathbb{R}^{n\times n}$ and for any $x \in U$, we next prove that:

$$\min_{i,j} M_{ij} \leq x^T Mx \leq \max_{i,j} M_{ij}, \forall x \in U. \tag{8}$$

If fact, let $(Mx)_i$ denote the i-th element of the vector Mx. Then

$$x^T Mx = \sum_{i=1}^{n} x_i (Mx)_i$$

But, $(Mx)_i$ is bounded by

$$\min\{\sum_{j=1}^{n} M_{ij}x_j : x \in U\} \leq (Mx)_i \leq \max\{\sum_{j=1}^{n} M_{ij}x_j : x \in U\}, \forall x \in U.$$

Since the linear programs $\min\{\sum_{j=1}^{n} M_{ij}x_j : x \in U\}$ and $\max\{\sum_{j=1}^{n} M_{ij}x_j : x \in U\}$ have optimal solutions on vertices, the optimal values of the above linear programs are exactly $\min_j\{M_{ij}\}$ and $\max_j\{M_{ij}\}$. Hence, we can compute bounds for $x^T Mx$ on U as follows:

$$\min_{i,j}\{M_{ij}\} = \min\{\sum_i x_i \min_j\{M_{ij}\} : x \in U\} \leq \sum_i x_i \min_j\{M_{ij}\} \leq \sum_i x_i (Mx)_i$$

$$= x^T Mx \leq \sum_i x_i \max_j\{M_{ij}\} \leq \max\{\sum_i x_i \max_j\{M_{ij}\} : x \in U\} = \max_{i,j}\{M_{i,j}\}.$$

Hence, (8) is true.
Using the bounds (8) for the matrices B and C, we have:

$$\min_{i,j}\{-B_{ij}\} \leq -x^T Bx \leq \max_{i,j}\{-B_{ij}\},$$

$$\min_{i,j}\{-C_{ij}\} \leq -x^T Cx \leq \max_{i,j}\{-C_{ij}\}$$

Since $A \in$PD, we have

$$0 < s = \min\{x^T Ax : x \in U\} \leq x^T Ax \leq \max_{i,j}\{A_{ij}\}, \forall x \in U.$$

Accordingly, $\frac{-x^T Bx}{2x^T Ax}$ is bounded by:

$$\frac{\min\{-B_{ij}\}}{2x^T Ax} \leq \frac{-x^T Bx}{2x^T Ax} \leq \frac{\max\{-B_{ij}\}}{2x^T Ax} \leq \gamma = \begin{cases} \frac{\max\{-B_{ij}\}}{2s}, \text{ if } \max\{-B_{ij}\} > 0; \\ \frac{\max\{-B_{ij}\}}{2\max\{A_{ij}\}}, \text{ if } \max\{-B_{ij}\} \leq 0. \end{cases}$$

and

$$\frac{\min\{-B_{ij}\}}{2x^T Ax} \geq \beta = \begin{cases} \frac{\min\{-B_{ij}\}}{2\max\{A_{ij}\}}, \text{ if } \min\{-B_{ij}\} > 0; \\ \frac{\min\{-B_{ij}\}}{2s}, \text{ if } \min\{-B_{ij}\} \leq 0. \end{cases}$$

Then

$$(\frac{-x^TBx}{2x^TAx})^2 + \frac{-x^TCx}{x^TAx} \leq \max\{\gamma^2, \beta^2\} + \frac{\max\{-C_{ij}\}}{s} = \alpha.$$

Finally, we can compute bounds for λ as follows:

$$\beta - \sqrt{\alpha} \leq \frac{-x^TBx}{2x^TAx} - \sqrt{(\frac{-x^TBx}{2x^TAx})^2 + \frac{-x^TCx}{x^TAx}} \leq$$

$$\lambda \leq \frac{-x^TBx}{2x^TAx} + \sqrt{(\frac{-x^TBx}{2x^TAx})^2 + \frac{-x^TCx}{x^TAx}} \leq \gamma + \sqrt{\alpha}.$$

□

In practice, it is interesting to compare in the future the bound proposed here with the one given in [6]. The bounds given in this paper have been designed such that they can be computed in a small amount of effort, even for large-scale problems.

4 DC Algorithms for solving P_{DC} and P'_{DC}

In this section, we investigate how to solve the DC programming formulations (P_{DC}) and (P'_{DC}).

Given a general DC program:

$$\min\{g(x) - h(x) : x \in C\},$$

where C is a non-empty convex set, the general DC algorithm (DCA) consists of constructing two sequences $\{x^k\}$ and $\{y^k\}$ via the following scheme[12–14]:

$$\begin{array}{c} x^k \to y^k \in \partial h(x^k) \\ \swarrow \\ x^{k+1} \in \partial g^*(y^k) = \operatorname{argmin}\{g(x) - \langle x, y^k \rangle : x \in C\}. \end{array}$$

The symbol ∂h stands for the sub-differential of the convex function h, and g^* is the conjugate function of g. These definitions are fundamental and can be found in any textbook of the convex analysis (see for example [17]).

The sequence $\{x^k\}$ and $\{y^k\}$ are respectively candidates for optimal solutions of the primal and dual DC programs.

In DCA, two major computations should be considered:

1. Computing $\partial h(x^k)$ to get y^k.
2. Solving the convex program $\operatorname{argmin}\{g(x) - \langle x, y^k \rangle : x \in C\}$ to obtain x^{k+1}.

Now, we investigate the use of DCA to solve the DC programs (P_{DC}) and (P'_{DC}). Concerning to (P_{DC}), since the function h is differentiable, $\partial h(x, y, z, w, \lambda)$ is reduced to a singleton $\{\nabla h(x,y,z,w,\lambda)\}$, where

$$\nabla h(x,y,z,w,\lambda) = \nabla g(x,y,z,w,\lambda) - \nabla f(x,y,z,w,\lambda) = \begin{bmatrix} \frac{x+w}{2} + (\rho_1+\rho_2)x + 2\lambda y - 2\lambda^2 x - w \\ (\rho_1+\rho_2-2\lambda^2)y + 2\lambda(x+z) \\ \rho_1 z + 2\lambda y \\ \frac{w-x}{2} \\ (\rho_1+\rho_2-2(\|x\|^2+\|y\|^2))\lambda + 2y^T(x+z) \end{bmatrix}. \tag{9}$$

For (P'_{DC}), since the function h' is non-differentiable, we compute the convex set $\partial h'(x,y,z,w,\lambda)$ as follows:

$$\partial h'(x,y,z,w,\lambda) = \left\{ \begin{bmatrix} (\rho_1+\rho_2)x + 2\lambda y - 2\lambda^2 x - u \\ (\rho_1+\rho_2-2\lambda^2)y + 2\lambda(x+z) \\ \rho_1 z + 2\lambda y \\ -v \\ (\rho_1+\rho_2-2(\|x\|^2+\|y\|^2))\lambda + 2y^T(x+z) \end{bmatrix} \right\} \tag{10}$$

where

$$u = (u_i)_{i=1,\dots,n}, u_i = \begin{cases} 1, & x_i < w_i; \\ \{0,1\}, & x_i = w_i; \\ 0, & x_i > w_i. \end{cases}$$

$$v = (v_i)_{i=1,\dots,n}, v_i = \begin{cases} 0, & x_i < w_i; \\ \{0,1\}, & x_i = w_i; \\ 1, & x_i > w_i. \end{cases}$$

Finally, DCA applied to (P_{DC}) and (P'_{DC}) requires solving respectively one convex quadratic program over a polyhedral convex set in each iteration.

The following two fixed-point schemes describe our dc algorithms:

$$\begin{array}{l}(x^{k+1},y^{k+1},z^{k+1},w^{k+1},\lambda^{k+1}) = \operatorname{argmin}\{g(x,y,z,w,\lambda) \\ -\langle (x,y,z,w,\lambda), \nabla h(x^k,y^k,z^k,w^k,\lambda^k)\rangle : (x,y,z,w,\lambda) \in \mathcal{C}\}\end{array} \tag{11}$$

with $g(x,y,z,w,\lambda) = \frac{\|x+w\|^2}{4} + \frac{\rho_1+\rho_2}{2}\|x\|^2 + (\frac{\rho_1+\rho_2}{2}+1)\|y\|^2 + (\frac{\rho_1}{2}+1)\|z\|^2 + \frac{\rho_1+\rho_2}{2}\lambda^2$.

$$\begin{array}{l}(x^{k+1},y^{k+1},z^{k+1},w^{k+1},\lambda^{k+1}) = \operatorname{argmin}\{g'(x,y,z,\lambda) \\ -\langle (x,y,z,w,\lambda), Y^k\rangle : (x,y,z,w,\lambda) \in \mathcal{C}\}\end{array} \tag{12}$$

with $Y^k \in \partial h'(x^k,y^k,z^k,w^k,\lambda^k)$ and $g'(x,y,z,\lambda) = \frac{\rho_1+\rho_2}{2}\|x\|^2 + (\frac{\rho_1+\rho_2}{2} + 1)\|y\|^2 + (\frac{\rho_1}{2}+1)\|z\|^2 + \frac{\rho_1+\rho_2}{2}\lambda^2$.

These convex quadratic programs can be efficiently solved via a quadratic programming solver such as CPLEX, Gurobi, XPress, etc.

DCA should terminate if one of the following stopping criteria is satisfied for given tolerances ϵ_1, ϵ_2 and ϵ_3.

(1) The sequence $\{(x^k, y^k, z^k, w^k, \lambda^k)\}$ converges, i.e.,

$$\|(x^{k+1}, y^{k+1}, z^{k+1}, w^{k+1}, \lambda^{k+1}) - (x^k, y^k, z^k, w^k, \lambda^k)\| \leq \epsilon_1$$

(2) The sequence $\{f(x^k, y^k, z^k, w^k, \lambda^k)\}$ (resp. $\{f'(x^k, y^k, z^k, w^k, \lambda^k)\}$) converges, i.e.,

$$\|f(x^{k+1}, y^{k+1}, z^{k+1}, w^{k+1}, \lambda^{k+1}) - f(x^k, y^k, z^k, w^k, \lambda^k)\| \leq \epsilon_2$$

$$(\text{resp. } \|f'(x^{k+1}, y^{k+1}, z^{k+1}, w^{k+1}, \lambda^{k+1}) - f'(x^k, y^k, z^k, w^k, \lambda^k)\| \leq \epsilon_2).$$

(3) *The sufficient global ϵ-optimality condition* holds, i.e.,

$$f(x^k, y^k, z^k, w^k, \lambda^k) \leq \epsilon_3 \quad (\text{resp. } f'(x^k, y^k, z^k, w^k, \lambda^k) \leq \epsilon_3).$$

The following theorem indicates the convergence of DCA:

Theorem 1 (Convergence theorem of DCA). *DCA applied to QEiCP generates convergence sequences $\{(x^k, y^k, z^k, w^k, \lambda^k)\}$ and $\{f(x^k, y^k, z^k, w^k, \lambda^k)\}$ (resp. $\{f'(x^k, y^k, z^k, w^k, \lambda^k)\}$) such that:*

- *The sequence $\{f(x^k, y^k, z^k, w^k, \lambda^k)\}$ (resp. $\{f'(x^k, y^k, z^k, w^k, \lambda^k)\}$) is decreasing and bounded below.*
- *The sequence $\{(x^k, y^k, z^k, w^k, \lambda^k)\}$ converges either to a solution of QEiCP when the third stopping condition is satisfied or to a general KKT point of (P_{DC}) (resp. (P'_{DC})).*

Proof. The proof of the theorem is an obvious consequence of the general convergence theorem of DCA [12–14]. The sufficient global optimality condition is due to the fact that the optimal value of the dc program is equal to zero. □

5 Conclusions

In this paper, we have presented two DC programming formulations of the Quadratic Eigenvalue Complementarity Problem. The corresponding numerical solution algorithms based on the classical DCA for solving these dc programs were briefly discussed.

The numerical results and the analysis of the performance of DCA for solving QEiCP will be given in a future paper. We will discuss a new *local dc decomposition algorithm* that is designed to speed up the convergence of DCA. Furthermore, that paper will also be devoted to the solution of QEiCP when the condition $A \in$PD and $C \notin S_0$ holds. A new DC formulation of QEiCP based on the reformulation of an equivalent extended EiCP will be introduced to deal with this case and the corresponding DC Algorithm will be discussed.

Acknowledgements

The research of Yi-Shuai Niu in this project is partially supported and financed by the Innovative Research Fund of Shanghai Jiao Tong University 985 Program. Joaquim Júdice was partially supported in the scope of R&D Unit UID/EEA/5008/2013, financed by the applicable financial framework (FCT/MEC through national funds and the applicable co-funded by FEDER-PT2002 partnership agreement).

References

1. Adly, S., Seeger, A. A non-smooth algorithm for cone constrained eigenvalue problems. Computational Optimization and Applications. Vol. 49, 299–318 (2011).
2. Adly, S., Rammal, H. A new method for solving second-order cone eigenvalue complementarity problem. Journal of Optimization Theory and Applications. (2014), doi:10.1007/s10957-014-0645-0.
3. Brás, C, Fukushima, M.. Júdice, J., Rosa, S. Variational inequality formulation for the asymmetric eigenvalue complementarity problem and its solution by means of a gap function. Pacific Journal of Optimization. Vol. 8, 197–215 (2012).
4. Brás, C, Iusem, A.N.. Júdice, J. On the quadratic eigenvalue complementarity problem. To appear in Journal of Global Optimization.
5. Fernandes, L. M., Júdice, J., Fukushima, M., Iusem, A. On the symmetric quadratic eigenvalue complementarity problem. Optimization Methods and Software, Vol. 29, 751–770 (2014).
6. Fernandes, L.M., Júdice, J., Sherali, H.D., Forjaz M.A., On an enumerative algorithm for solving eigenvalue complementarity problems. Computational Optimization and Applications. Vol. 59, 113–134 (2014).
7. Júdice, J, Sherali, H. D., Ribeiro, I. The eigenvalue complementarity problem. Computational Optimization and Applications. Vol. 37, 139–156 (2007).
8. Júdice, J., Raydan, M., Rosa, S., Santos, S. On the solution of the symmetric complementarity problem by the spectral projected gradient method. Numerical Algorithms. Vol. 44, 391–407 (2008).
9. Júdice, J., Sherali, H. D., Ribeiro, I., Rosa, S. On the asymmetric eigenvalue complementarity problem. Optimization Methods and Software. Vol. 24, 549–586 (2009).
10. Le Thi, H.A., Moeini, M., Pham, D.T., Judice, J. A DC programming approach for solving the symmetric Eigenvalue Complementarity Problem, Computational Optimization and Applications, Vol. 51, 1097–1117 (2012).
11. Niu, Y.S., Le Thi, H.A., Pham, D.T., Júdice, J., Efficient dc programming approaches for the asymmetric eigenvalue complementarity problem. Optimization Methods and Software. Vol. 28, 812–829 (2013).
12. Pham Dinh, T. , Le Thi, H.A.: DC optimization algorithms for solving the trust region subproblem, SIAM Journal of Optimization, Vol. 8, 476–507 (1998).
13. Pham Dinh, T. , Le Thi, H.A.: DC Programming. Theory, Algorithms, Applications: The State of the Art. First International Whorkshop on Global Constrained Optimization and Constraint Satisfaction, Nice, October 2–4 (2002).
14. Pham Dinh, T. , Le Thi, H.A.: The DC programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems. Annals of Operations Research, Vol. 133, 23–46 (2005).

15. Pinto da Costa, A., Seeger, A. Cone constrained eigenvalue problems, theory and algorithms. Computational Optimization and Applications. Vol. 45, 25–57 (2010).
16. Queiroz, M.. Júdice, J.. Humes, C. The symmetric eigenvalue complementarity problem. Mathematics of Computation. Vol. 73, 1849–1863 (2003).
17. Rockafellar, R. Tyrell. Convex Analysis. Princeton: Princeton University Press (1970).
18. Seeger, A. Eigenvalue analysis of equilibrium processes defined by linear complementaritv conditions. Linear Algebra and Its Applications. Vol. 294, 1–14 (1999).
19. Seeger, A. Quadratic eigenvalue problems under conic constraints. SIAM Journal on Matrix Analysis and Applications. Vol. 32, 700–721 (2011).