*Optimization Methods and Software*
Vol. 00, No. 00, January 2010, 1–17

**RESEARCH ARTICLE**

# Efficient DC Programming Approaches for the Asymmetric Eigenvalue Complementarity Problem

Y.S. NIU[a], T. PHAM DINH[b], H.A. LE THI[c] and J.J. JUDICE[d]

[a,b]*National Institute of Applied Sciences, Rouen, France*; [c]*University of Paul Verlaine, Metz, France*; [d]*Universidade de Coimbra and Instituto de Telecomunicações, Portugal*

(*v1.0 released December 2010*)

In this paper, we propose nonlinear programming formulations (NLP) and DC (Difference of Convex functions) programming approaches for the asymmetric eigenvalue complementarity problem (EiCP). The EiCP has a solution if and only if these NLPs have zero global optimal value. We reformulate the NLPs as DC Programs (DCP) which can be efficiently solved by DCA (DC Algorithm). Some preliminary numerical results illustrate the good performance of the proposed methods.

**Keywords:** DC Programming; DCA; Complementarity Problems; Eigenvalue Problems; Nonlinear Programming.

## 1. Introduction

Given an asymmetric matrix $A \in \mathbb{R}^{n\times n}$ and a positive definite matrix $B \in S_n^{++}$ ( where $S_n^{++}$ denotes the set of all $n \times n$ symmetric positive definite matrices), the Asymmetric Eigenvalue Complementarity Problem (EiCP) consists of finding a real number $\lambda > 0$ (complementary eigenvalue) and a vector $x \in \mathbb{R}^n \setminus \{0\}$ (complementary eigenvector) such that:

$$(EiCP) \qquad \begin{cases} w = (\lambda B - A)x \\ w \geq 0, x \geq 0 \\ x^T w = 0. \end{cases}$$

This problem is very important both on the theoretical aspects of mathematics and in application fields (see [6, 7, 16, 20, 21, 23] and references therein). It was proved that the number of solutions for EiCP is finite and at most equal to $2^{n+1} - n - 2$ [6]. Solving the EiCP is in general a NP-hard problem since determining the feasibility of EiCP is already proved to be a NP-complete problem [6]. The class of the matrix $A$ plays a very important role in the solution of EiCP. It was shown that EiCP can be reduced to the problem of finding a stationary point of Rayleigh function on the simplex when $A$ and $B$ are both symmetric matrices [18, 22]. A DC programming approach for solving the symmetric case has been investigated in [8]. However, this result is no longer valid when the matrix $A$ is asymmetric. A

The paper has been presented in ICOTA8.

a NIU Yi-Shuai. Email: niuys@insa-rouen.fr
b PHAM DINH Tao. Email: pham@insa-rouen.fr
c LE THI Hoai An. Email: lethi@univ-metz.fr
d JUDICE Joaquim João. Email: Joaquim.Judice@co.it.pt

ISSN: 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/1055678xxxxxxxxxxxxx
http://www.informaworld.com

number of algorithms have been designed for solving the EiCP in this latter case [1, 6, 7, 17]. In particular, it has been shown in [7] that the EiCP is equivalent to the Nonlinear Program:

$$
(NLP) \qquad \begin{aligned} \min\ & f(x,y,w,z) = (y-zx)^T(y-zx) + x^T w \\ s.t.\ & w = Bx - Ay \\ & e^T x = 1 \\ & e^T y = z \\ & (x,y,w,z) \geq 0 \end{aligned}
$$

where $e$ denotes the all-ones vector. In fact, the EiCP has a solution $(\lambda^*, x^*)$ if and only if $(x^*, y^* = z^* x^*, w^*, z^* = \frac{1}{\lambda^*})$ is a solution of NLP with $f(x^*, y^*, w^*, z^*) = 0$. The objective function $f$ is a 4th order nonconvex polynomial function and the constraint set of NLP is a polyhedral convex set. So NLP is a nonlinear and nonconvex polynomial optimization program which is a NP-hard problem and difficult to be solved.

Our work will focus on proposing some new nonlinear programming formulations for the asymmetric EiCP and investigating how to reformulate these problems as DC programs and then solving them via efficient DC programming approaches - DCA. Some numerical results show a good performance of our approaches, especially their high-efficiency for tackling large-scale problems.

## 2. Nonlinear programming formulations

The asymmetric EiCP consists of finding $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n$ such that

$$
\begin{cases} v = (\lambda B - A)x \\ x^T v = 0 \\ 0 \neq x \geq 0, v \geq 0, \lambda > 0 \end{cases} \tag{1}
$$

where $A \in \mathbb{R}^{n \times n}$ is an asymmetric matrix, and $B \in S_n^{++}$.
If $z = \frac{1}{\lambda} > 0$, we have

$$
zv = Bx - zAx.
$$

Let $y = zx$ and $w = zv$. An equivalent formulation of EiCP is

$$
\begin{cases} w = Bx - Ay \\ y = zx \\ x^T w = 0 \\ 0 \neq x \geq 0, w \geq 0, z > 0. \end{cases} \tag{2}
$$

To avoid $x = 0$, we can restrict $x$ to the simplex $\{e^T x = 1, x \geq 0\}$ and obtain

$$
\begin{cases} w = Bx - Ay \\ e^T x = 1 \\ y = zx \\ x^T w = 0 \\ x \geq 0, w \geq 0, z > 0. \end{cases} \tag{3}
$$

Theoretically, if the EiCP has a solution, then there must exist a lower bound $l > 0$ and an upper bound $u > 0$ such that $0 < l \leq z \leq u$ for any solution of the EiCP and the constraint $z > 0$ in (3) can be replaced by $l \leq z \leq u$.

In practice, these lower and upper bounds for $z$ are not given beforehand. We can estimate the upper bound $u$ of $z$ by solving the linear program:

$$\begin{aligned} u = {} & \max z \\ s.t.\ \ & Bx - Ay \geq 0 \\ & e^T x = 1 \\ & e^T y = z \\ & x \geq 0, y \geq 0, z \geq 0. \end{aligned} \tag{4}$$

Furthermore, the following result provides a lower bound $l$ for $z$.

THEOREM 2.1 *Let $D = \frac{A+A^T}{2}$, and $\lambda_{min}(C)$, $\lambda_{max}(C)$ denote the smallest and the largest eigenvalue of the matrix $C$. If the EiCP has a solution, then $\lambda_{max}(D) > 0$ and*

$$z \geq \frac{\lambda_{min}(B)}{\lambda_{max}(D)} = l.$$

*Proof* Since in the formulation (3) $x^T w = 0, w = Bx - Ay, y = zx$, we get

$$x^T w = x^T(Bx - zAx) = x^T(B - zA)x = 0;$$

And $x^T Ax = (x^T Ax + x^T A^T x)/2 = x^T Dx$, we have

$$x^T(B - zA)x = x^T(B - zD)x = 0.$$

Note that $B - zD$ is a real symmetric matrix, because $z \in \mathbb{R}$, $B$ and $D = \frac{A+A^T}{2}$ are real symmetric matrices. Hence all eigenvalues of the symmetric matrix $B - zD$ are real. Moreover, $x^T(B - zD)x = 0$ implies that

$$\lambda_{min}(B - zD) \leq 0.$$

We then deduce that

$$\begin{aligned} 0 \geq \lambda_{min}(B - zD) &= \min\{x^T(B - zD)x : \|x\| = 1\} \\ &\geq \min\{x^T Bx : \|x\| = 1\} + z \min\{x^T(-D)x : \|x\| = 1\} \\ &= \lambda_{min}(B) - z\lambda_{max}(D). \end{aligned}$$

Since $\lambda_{min}(B) > 0$ and $z > 0$, $\lambda_{max}(D) \leq 0$ implies $0 \geq \lambda_{min}(B) - z\lambda_{max}(D) > 0$. This contradiction means that there is no feasible solution when $\lambda_{max}(D) \leq 0$. Therefore $\lambda_{max}(D) > 0$ is a necessary condition for the existence of solution of EiCP and

$$0 \geq \lambda_{min}(B) - z\lambda_{max}(D); \lambda_{max}(D) > 0 \Longrightarrow z \geq \frac{\lambda_{min}(B)}{\lambda_{max}(D)}.$$

∎

PROPOSITION 2.2 *If $l \leq z \leq u$, the solution $(x, y, w, z)$ of the EiCP satisfies*

$$x \in [0,1]^n; y \in [0,u]^n;$$

$$0 \leq w = Bx - Ay \leq \begin{bmatrix} \sum_j (|B_{1j}| + u|A_{1j}|) \\ \vdots \\ \sum_j (|B_{nj}| + u|A_{nj}|) \end{bmatrix}.$$

*Proof* For the bounds of $x$, we have

$$\{e^T x = 1, x \geq 0\} \Rightarrow x \in [0,1]^n.$$

As far as the bounds of $y$ are concerned, we have $\{y = zx, e^T x = 1, x \geq 0, z > 0\} \Rightarrow \{e^T y = ze^T x = z > 0, y = zx \geq 0\} \Rightarrow y \in [0,z]^n$. Hence $y \in [0,u]^n$.
The bounds of $w$ can be derived from $x \in [0,1]^n$, $y \in [0,u]^n$ and $w = Bx - Ay$ as

$$0 \leq w = Bx - Ay \leq \begin{bmatrix} \sum_j (|B_{1j}| + u|A_{1j}|) \\ \vdots \\ \sum_j (|B_{nj}| + u|A_{nj}|) \end{bmatrix}.$$

■

### 2.1. *Nonlinear Programming Formulation 1 (NLP1)*

We can penalize the nonlinear constraints $y = zx$ and $x^T w = 0$ of (3) to obtain the nonlinear program:

$$\begin{array}{ll} min & f(x,y,w,z) = ||y - zx||^2 + x^T w \\ s.t. & w = Bx - Ay \\ & e^T x = 1 \\ & x \geq 0, y \geq 0, w \geq 0, z > 0. \end{array} \tag{5}$$

If the optimal value of (5) is zero then (3) and (5) have the same solution set. According to theorem 2.1 and proposition 2.2, we can replace $z > 0$ by $l \leq z$ with $l = \frac{\lambda_{min}(B)}{\lambda_{max}(D)}$. This yields the following equivalent formulation of (5):

$$\begin{array}{ll} 0 = min & ||y - zx||^2 + x^T w \\ s.t. & w = Bx - Ay \\ & e^T x = 1 \\ & x \geq 0, y \geq 0, w \geq 0, z \geq l \end{array} \tag{6}$$

and all variables of (6) could be bounded as

$$l \leq z \leq u; x \in [0,1]^n; y \in [0,u]^n;$$

$$0 \le w \le \begin{bmatrix} \sum_j (|B_{1j}| + u|A_{1j}|) \\ \vdots \\ \sum_j (|B_{nj}| + u|A_{nj}|) \end{bmatrix}.$$

with $u$ being the optimal value of the problem (4). If (4) is an unbounded problem, then we can fix an upper bound $u$ and find solutions in the interval $l \le z \le u$.

The boundeness of the variables is important since we need these bounds for deriving a DC programming formulation for the problem (6). This topic will be investigated in the section 3.

### 2.2. *Nonlinear Programming Formulation 2 (NLP2)*

The NLP formulation (6) can be presented as

$$0 = \min_{(x,y,w)\in C} \min_{z\ge l} ||y - zx||^2 + x^T w \tag{7}$$

where $l := \frac{\lambda_{min}(B)}{\lambda_{max}(D)} > 0$ and $C := \{w = Bx - Ay, e^T x = 1, x \ge 0, y \ge 0, w \ge 0\}$ is a polyhedral convex set. It is easy to prove that if the problem (7) has a zero optimal value then its optimal solution set is the same as the solution set of (3).

Note that we can replace $z \ge l$ by $z \ge 0$ since by theorem 2.1 there is no zero optimal solution in the interval $0 \le z < l$. Therefore, we have

$$\min_{(x,y,w)\in C} \min_{z\ge 0} ||y - zx||^2 + x^T w = \min_{(x,y,w)\in C} \{||y||^2 + x^T w + \min_{z\ge 0}\{z^2||x||^2 - (2x^T y)z\}\}.$$

Let $\phi_{x,y}(z) := z^2||x||^2 - (2x^T y)z$, we have

$$\phi'_{x,y}(z) = 2z||x||^2 - (2x^T y) = 0 \Longrightarrow z = \frac{x^T y}{||x||^2} \ge 0, \forall (x, y, w) \in C.$$

Hence,

$$\min_{z\ge 0} \phi_{x,y}(z) = -\frac{(x^T y)^2}{||x||^2}, \forall (x, y, w) \in C.$$

The problem (7) is equivalent to

$$0 = \min_{(x,y,w)\in C} ||y||^2 + x^T w - \frac{(x^T y)^2}{||x||^2}. \tag{8}$$

The objective function of (8) is nonlinear and nonconvex. Hence the problem (8) is a nonlinear and nonconvex program. Its DC decomposition and solution method will be discussed in section 3.

### 2.3. *Nonlinear Programming Formulation 3 (NLP3)*

In the proof of the theorem 2.1, we showed that $x^T w = x^T(B - zD)x$. By using this equality in the formulation (6), we can derive the equivalent formulation:

$$\begin{array}{ll} 0 = min & ||y - zx||^2 + x^T(B - zD)x \\ s.t. & Bx - Ay \geq 0 \\ & e^T x = 1 \\ & x \geq 0, y \geq 0, z \geq l. \end{array} \tag{9}$$

Obviously, if the optimal value of (9) is zero then its optimal solution set is the same as that of (3). Replacing $z \geq l$ by $z \geq 0$, the problem (9) can be represented by

$$0 = \min_{(x,y)\in C_1} \min_{z\geq 0} ||y - zx||^2 + x^T(B - zD)x \tag{10}$$

where $C_1 := \{Bx - Ay \geq 0, e^T x = 1, x \geq 0, y \geq 0\}$.

$$\min_{(x,y)\in C_1} \min_{z\geq 0} ||y-zx||^2+x^T(B-zD)x = \min_{(x,y)\in C_1} \{||y||^2+x^T Bx+\min_{z\geq 0}\{z^2||x||^2-(2x^T y+x^T Dx)z\}\}.$$

Let $\phi_{x,y}(z) := z^2||x||^2 - (2x^T y + x^T Dx)z$, we deduce that

$$\phi'_{x,y}(z) = 2z||x||^2 - (2x^T y + x^T Dx) = 0 \Longrightarrow z = \frac{x^T Dx + 2x^T y}{2||x||^2},$$

and

$$\min_{z\geq 0} \phi_{x,y}(z) = \begin{cases} -\frac{(x^T Dx+2x^T y)^2}{4||x||^2}, & x^T Dx + 2x^T y > 0; \\ 0, & x^T Dx + 2x^T y \leq 0. \end{cases}$$

Proposition 2.3 *If (10) holds, then* $x^T Dx + 2x^T y > 0$.

*Proof* When $x^T Dx + 2x^T y \leq 0$, we have $\min_{z\geq 0}\{\phi_{x,y}(z)\} = 0$ and the problem (10) becomes

$$\min_{(x,y)\in C_1} ||y||^2 + x^T Bx.$$

Since $B$ is a PD matrix, this problem has a unique optimal solution $(x = 0, y = 0)$. But $(0, 0) \notin C_1$ and this establishes the theorem. ■

*Remark 1* If $A$ is a copositive matrix ($A \in \mathcal{CO}_n$), i.e. $x^T Ax \geq 0$ for all $x \geq 0$, then $D = \frac{A+A^T}{2} \in \mathcal{CO}_n$ and we have $x^T Dx + 2x^T y \geq 0, \forall(x, y) \in C_1$.

The EiCP has a solution if and only if

$$\begin{array}{ll} 0 = min & ||y||^2 + x^T Bx - \frac{(x^T Dx+2x^T y)^2}{4||x||^2} \\ s.t. & (x, y) \in C_1 \\ & x^T Dx + 2x^T y > 0. \end{array} \tag{11}$$

## 3. DC Program Formulations and DCA

This section will focus on how to reformulate the nonlinear programs presented in section 2 as DC programs and to propose DC Algorithms (DCA) for their numerical solutions. DCA was first introduced by Pham Dinh Tao in their preliminary form in 1985. It has been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao (see e.g. [11–14] and also the web page `http://lita.sciences.univ-metz.fr/~lethi/`). DCA has been successfully applied to many large-scale (smooth or nonsmooth) nonconvex programs in different fields of applied sciences for which they often give global optimal solutions. The algorithm has proved to be usually more robust and efficient than standard methods.

For DC decomposition, we should represent the nonconvex objective function as a DC function. In general, there exists infinitely many DC decompositions for a nonconvex function. In fact, if $f$ has a DC decomposition as $g - h$, then for any convex function $p$, $(g + p) - (h + p)$ is also a DC decomposition.

### 3.1. *DC Formulation and DCA for NLP1*

#### *3.1.1. DC formulation for NLP1*

We rewrite the objective function of NLP1 as follows:

$$f(x, y, w, z) = ||y-zx||^2+x^T w = ||y||^2+w^T x-2zx^T y+z^2||x||^2 = f_0(y)+f_1(x, w)+f_2(x, y, z)+f_3(x, z)$$

with

$$f_0(y) = ||y||^2; f_1(x, w) = w^T x; f_2(x, y, z) = -2zx^T y; f_3(x, z) = z^2||x||^2.$$

Then $f_0$ is a convex quadratic function and $f_1, f_2, f_3$ are nonconvex polynomial functions. The gradients of these nonconvex functions are:

$$\begin{cases} \nabla_x f_1(x, w) = w \\ \nabla_w f_1(x, w) = x \end{cases},$$

$$\begin{cases} \nabla_x f_2(x, y, z) = -2zy \\ \nabla_y f_2(x, y, z) = -2zx \\ \nabla_z f_2(x, y, z) = -2x^T y \end{cases}$$

and

$$\begin{cases} \nabla_x f_3(x, z) = 2z^2 x \\ \nabla_z f_3(x, z) = 2z||x||^2 \end{cases}.$$

The Hessian matrices are

$$\nabla^2 f_1(x, w) = \begin{bmatrix} O & I \\ I & O \end{bmatrix}, \text{ a } 2n \times 2n \text{ matrix},$$

$$\nabla^2 f_2(x,w) = \begin{bmatrix} O & -2zI & -2y \\ -2zI & O & -2x \\ -2y^T & -2x^T & O \end{bmatrix}, \text{ a } (2n+1)\times(2n+1) \text{ matrix,}$$

and

$$\nabla^2 f_3(x,z) = \begin{bmatrix} 2z^2 I & 4zx \\ 4zx^T & 2x^T x \end{bmatrix}, \text{ a } (n+1)\times(n+1) \text{ matrix.}$$

We can estimate the upper bounds of their spectral radius as

$$\rho(\nabla^2 f_1(x,w)) = 1 = \rho_1.$$

$$\begin{aligned}\rho(\nabla^2 f_2(x,y,z)) &\leq \|\nabla^2 f_2(x,y,z)\|_1 = \max_{i=1,\dots,n}\{2z+2y_i, 2z+2x_i, 2e^T y + 2e^T x\} \\ &= \max_{i=1,\dots,n}\{2z+2y_i, 2z+2x_i, 2z+2\} \leq \max\{4z, 2z+2\} \\ &\leq \max\{4u, 2u+2\} = \rho_2.\end{aligned}$$

$$\begin{aligned}\rho(\nabla^2 f_3(x,z)) &\leq \|\nabla^2 f_3(x,z)\|_1 = \max_{i=1,\dots,n}\{2z^2+4zx_i, 4ze^T x + 2||x||^2\} \\ &\leq \max\{2z^2+4z, 4z+2\} \leq \max\{2u^2+4u, 4u+2\} = \rho_3.\end{aligned}$$

For the bilinear function $f_1(x,w) = x^T w$, we have two types of DC decompositions:

(1) $f_1(x,w) = x^T w = (\frac{||x+w||^2}{4}) - (\frac{||x-w||^2}{4})$.
(2) $f_1(x,w) = x^T w = (\frac{\rho_1 ||(x,w)||^2}{2}) - (\frac{\rho_1 ||(x,w)||^2}{2} - f_1(x,w))$
$= (\frac{||x||^2+||w||^2}{2}) - (\frac{||x-w||^2}{2})$.

So two DC decompositions can be defined

(i) $f(x,y,w,z) = g_1(x,y,w,z) - h_1(x,y,w,z)$ with

$$g_1(x,y,w,z) = \frac{\rho_1+\rho_2+\rho_3}{2}\|x\|^2 + (\frac{\rho_2}{2}+1)\|y\|^2 + \frac{\rho_1}{2}\|w\|^2 + \frac{\rho_2+\rho_3}{2}z^2,$$

$$h_1(x,y,w,z) = g_1(x,y,w,z) - f(x,y,w,z).$$

(ii) $f(x,y,w,z) = g_2(x,y,w,z) - h_2(x,y,w,z)$ with

$$g_2(x,y,w,z) = \frac{||x+w||^2}{4} + \frac{\rho_2+\rho_3}{2}\|x\|^2 + (\frac{\rho_2}{2}+1)\|y\|^2 + \frac{\rho_2+\rho_3}{2}z^2,$$

$$h_2(x,y,w,z) = g_2(x,y,w,z) - f(x,y,w,z).$$

#### *3.1.2. DCA for NLP1*

For the DC program defined by

$$(P_{DC}) \qquad \min\{g(x) - h(x) : x \in C\}$$

where $C$ is a nonempty convex set, DCA yields the following general scheme: Construct two sequences $\{x^k\}$ and $\{y^k\}$ ( candidates to be optimal solutions of the primal and dual DC programs resp.) as follows:

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

In our application, since $h_1$ and $h_2$ are both differentiable functions, we have $\partial h_1(x,y,w,z) = \{\nabla h_1(x,y,w,z)\}$ and $\partial h_2(x,y,w,z) = \{\nabla h_2(x,y,w,z)\}$ with

$$\nabla h_1(x,y,w,z) = \nabla g_1(x,y,w,z) - \nabla f(x,y,w,z) = \begin{bmatrix} (\rho_1+\rho_2+\rho_3)x - w + 2zy - 2z^2x \\ \rho_2 y + 2zx \\ \rho_1 w - x \\ (\rho_2+\rho_3)z + 2x^Ty - 2z\|x\|^2 \end{bmatrix}$$

and

$$\nabla h_2(x,y,w,z) = \nabla g_2(x,y,w,z) - \nabla f(x,y,w,z) = \begin{bmatrix} (\frac{1}{2}+\rho_2+\rho_3)x - \frac{1}{2}w + 2zy - 2z^2x \\ \rho_2 y + 2zx \\ \frac{w-x}{2} \\ (\rho_2+\rho_3)z + 2x^Ty - 2z\|x\|^2 \end{bmatrix}.$$

Finally, DCA for the two DC decompositions ($g_1 - h_1$ and $g_2 - h_2$) yields the same fixed point framework as:

$$(x^{k+1}, y^{k+1}, w^{k+1}, z^{k+1}) \in argmin\{g_i(x,y,w,z) - \langle (x,y,w,z), \nabla h_i(x^k,y^k,w^k,z^k)\rangle : (x,y,w,z) \in C\}$$

where $i = 1, 2$. We observed from the above framework that each iteration of DCA requires solving a strictly convex quadratic program over a convex polyhedral set $C$. This kind of problem can be efficiently solved by many quadratic programming solvers (such as CPLEX, MATLAB etc.).

For given tolerances $\epsilon_1, \epsilon_2$ and $\epsilon_3$, DCA could be terminated when one of the following conditions is satisfied:

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

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

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

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

(3) The sufficient global $\epsilon$-optimality condition:

$$|f(x^k, y^k, w^k, z^k)| \leq \epsilon_3.$$

The first two conditions are general stopping criteria for DCA. The third one is a special feature for this specific problem since we know what the global optimal value should be if EiCP has a solution.

THEOREM 3.1 *DCA has the following convergence theorem:*

- *DCA generates convergence sequences $\{(x^k, y^k, w^k, z^k)\}$ and $\{f(x^k, y^k, w^k, z^k)\}$, such that $\{f(x^k, y^k, w^k, z^k)\}$ is decreasing and bounded below.*
- *The sequence $\{(x^k, y^k, w^k, z^k)\}$ converges either to a feasible solution of EiCP when the third stopping criterion is satisfied or to a general KKT solution of NLP when the first and the second stopping criteria are satisfied.*

The proof of Theorem 3.1 is based on the general convergence theorem of DCA ( see [10–15] for details).

### 3.2. *DC Formulation and DCA for NLP2*

#### *3.2.1. DC formulation for NLP2*

The nonlinear objective function of NLP2 can be written as:

$$f(x, y, w) = ||y||^2 + x^T w - \frac{(x^T y)^2}{||x||^2} = f_0(y) + f_1(x, w) + f_2(x, y)$$

with

$$f_0(y) = ||y||^2; f_1(x, w) = w^T x; f_2(x, y) = -\frac{(x^T y)^2}{||x||^2}.$$

Then $f_0$ is a convex quadratic function, and $f_1, f_2$ are nonconvex functions. The gradients of the nonconvex functions are:

$$\begin{cases} \nabla_x f_1(x, w) = w \\ \nabla_w f_1(x, w) = x \end{cases}$$

$$\begin{cases} \nabla_x f_2(x, y) = -2\frac{x^T y}{||x||^2} y + 2\frac{(x^T y)^2}{||x||^2} x \\ \nabla_y f_2(x, y) = -2\frac{(x^T y)}{||x||^2} x \end{cases}.$$

The Hessian matrix of $f_1(x, w)$ and its spectral radius are as follows:

$$\nabla^2 f_1(x, w) = \begin{bmatrix} O & I \\ I & O \end{bmatrix}, \text{ a } 2n \times 2n \text{ matrix.}$$

$$\rho(\nabla^2 f_1(x, w)) = 1 = \rho_1.$$

The Hessian matrix of $f_2(x, w)$ is given by

$$\nabla^2 f_2(x, w) = \begin{bmatrix} \nabla^2_{x,x} f_2(x, y) & \nabla^2_{x,y} f_2(x, y) \\ \nabla^2_{y,x} f_2(x, y) & \nabla^2_{y,y} f_2(x, y) \end{bmatrix}, \text{ a } 2n \times 2n \text{ matrix.}$$

with

$$\begin{cases} \nabla^2_{x,x} f_2(x, y) = -\frac{2}{||x||^2}(yy^T) - \frac{8(x^T y)^2}{||x||^6}(xx^T) + \frac{2(x^T y)^2}{||x||^2} I_n + \frac{4x^T y}{||x||^4}(xy^T + yx^T) \\ (\nabla^2_{y,x} f_2(x, y))^T = \nabla^2_{x,y} f_2(x, y) = -\frac{2}{||x||^2}(yx^T) + \frac{4x^T y}{||x||^4}(xx^T) - \frac{2x^T y}{||x||^2} I_n \\ \nabla^2_{y,y} f_2(x, y) = -\frac{2}{||x||^2}(xx^T) \end{cases}.$$

For estimating the spectral radius of the Hessian matrix $\nabla^2 f_2(x,y)$, we need the following theorem.

THEOREM 3.2 *Let* $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$ *with all blocks* $A_i, i = 1,\dots,4$ *are* $n \times n$ *real matrices. Then*

$$\|A\|_2^2 \le \sum_{i=1}^{4} \|A_i\|_2^2.$$

*Proof* By definition

$$\|A\|_2^2 = \max_{(x,y)\neq 0} \frac{\left\| A \begin{bmatrix} x \\ y \end{bmatrix} \right\|_2^2}{\|x\|_2^2 + \|y\|_2^2}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. Hence

$$\begin{aligned} \left\| A \begin{bmatrix} x \\ y \end{bmatrix} \right\|_2^2 &= \left\| \begin{matrix} A_1x + A_2y \\ A_3x + A_4y \end{matrix} \right\|_2^2 = \|A_1x + A_2y\|_2^2 + \|A_3x + A_4y\|_2^2 \\ &\le (\|A_1x\|_2 + \|A_2y\|_2)^2 + (\|A_3x\|_2 + \|A_4y\|_2)^2 \\ &\le (\|A_1\|_2\|x\|_2 + \|A_2\|_2\|y\|_2)^2 + (\|A_3\|_2\|x\|_2 + \|A_4\|_2\|y\|_2)^2 \\ &\le (\|A_1\|_2^2 + \|A_2\|_2^2)(\|x\|_2^2 + \|y\|_2^2) + (\|A_3\|_2^2 + \|A_4\|_2^2)(\|x\|_2^2 + \|y\|_2^2) \\ &= (\|A_1\|_2^2 + \|A_2\|_2^2 + \|A_3\|_2^2 + \|A_4\|_2^2)(\|x\|_2^2 + \|y\|_2^2) \end{aligned}$$

and

$$\|A\|_2^2 = \max_{(x,y)\neq 0} \frac{\left\| A \begin{bmatrix} x \\ y \end{bmatrix} \right\|_2^2}{\|x\|_2^2 + \|y\|_2^2} \le \max_{(x,y)\neq 0} \frac{(\|x\|_2^2 + \|y\|_2^2) \sum_{i=1}^{4} \|A_i\|_2^2}{\|x\|_2^2 + \|y\|_2^2} = \sum_{i=1}^{4} \|A_i\|_2^2.$$

■

Using this theorem 3.2, we can prove the following result.

THEOREM 3.3

$$\|\nabla^2 f_2(x,y)\|_2^2 \le (2nu^2)^2(9n+1)^2 + 128n^2u^2 + 4$$

*for all solutions of the EiCP.*

*Proof* Let $\|.\|$ denote the matrix and vector 2-norms. According to the theorem 3.2, we get

$$\begin{aligned} \|\nabla^2 f_2(x,y)\|^2 &\le \|\nabla^2_{x,x} f_2(x,y)\|^2 + 2\|\nabla^2_{x,y} f_2(x,y)\|^2 + \|\nabla^2_{y,y} f_2(x,y)\|^2 \\ &\le (\|\tfrac{2}{\|x\|^2} yy^T\| + \|\tfrac{8(x^Ty)^2}{\|x\|^6} xx^T\| + \|\tfrac{2(x^Ty)^2}{\|x\|^2} I_n\| + \|\tfrac{4x^Ty}{\|x\|^4} xy^T\| + \|\tfrac{4x^Ty}{\|x\|^4} yx^T)\|)^2 \\ &\quad +2(\|\tfrac{2}{\|x\|^2} yx^T\| + \|\tfrac{4x^Ty}{\|x\|^4} xx^T\| + \|\tfrac{2|x^Ty|}{\|x\|^2} I_n\|)^2 + \|\tfrac{2}{\|x\|^2} xx^T\|^2. \end{aligned} \tag{12}$$

For any solution of EiCP, due to the proposition 2.2, we have $y \in [0,u]^n$ and $e^Tx = 1, x \ge 0$, which yields $\frac{1}{\sqrt{n}} \le \|x\| \le 1$ and $\|y\| \le u\sqrt{n}$. On the other hand, since $(xy^T)x = (y^Tx)x$, then $y^Tx$ is the only possible nonzero eigenvalue of the rank-one matrix $xy^T$ with eigenvector x. Moreover, we have $\|xy^T\| = \|x\|\|y\|$.

Now, we use this result to get upper-bounds for some of the norms in (12).

(1)

$$\|\frac{2}{||x||^2}xx^T\| = \frac{2}{||x||^2}\|xx^T\| = 2.$$

(2)

$$\|\frac{2|x^Ty|}{||x||^2}I_n\| = \frac{2|x^Ty|}{||x||^2} \leq \frac{2||x||||y||}{||x||^2} = \frac{2||y||}{||x||} \leq 2nu.$$

(3)

$$\|\frac{4x^Ty}{||x||^4}xx^T\| = \frac{4|x^Ty|}{||x||^2} \leq 4nu.$$

(4)

$$\|\frac{2}{||x||^2}yx^T\| = \frac{2||x||||y||}{||x||^2} \leq 2nu.$$

(5) $\frac{4x^Ty}{||x||^4}xy^T$ and $\frac{4x^Ty}{||x||^4}yx^T$ are rank-one matrices with the same only possible nonzero eigenvalue $\frac{4(x^Ty)^2}{||x||^4}$. Therefore,

$$\|\frac{4x^Ty}{||x||^4}xy^T\| = \|\frac{4x^Ty}{||x||^4}yx^T\| = \frac{4|x^Ty|||y||}{||x||^3} \leq \frac{4||y||^2}{||x||^2} \leq 4n^2u^2.$$

(6)

$$\|\frac{2(x^Ty)^2}{||x||^2}I_n\| = \frac{2(x^Ty)^2}{||x||^2} \leq \frac{2||x||^2||y||^2}{||x||^2} = 2||y||^2 \leq 2nu^2.$$

(7)

$$\|\frac{8(x^Ty)^2}{||x||^6}xx^T\| = \frac{8(x^Ty)^2}{||x||^4} \leq \frac{8||x||^2||y||^2}{||x||^4} = \frac{8||y||^2}{||x||^2} \leq 8n^2u^2.$$

(8)

$$\|\frac{2}{||x||^2}yy^T\| = \frac{2||y||^2}{||x||^2} \leq 2n^2u^2.$$

Finally, we get from (12) and (1)-(8) that

$$\begin{aligned}\|\nabla^2 f_2(x,y)\|^2 &\leq (2n^2u^2 + 8n^2u^2 + 2nu^2 + 4n^2u^2 + 4n^2u^2)^2 + 2(2un + 4un + 2un)^2 + 2^2 \\ &= (2nu^2)^2(9n+1)^2 + 128n^2u^2 + 4.\end{aligned}$$

■

According to the theorem 3.3, we found an upper bound for the spectral radius:

$$\rho(\nabla^2 f_2(x,y)) = \|\nabla^2 f_2(x,y)\|_2 \leq \sqrt{(2nu^2)^2(9n+1)^2 + 128n^2u^2 + 4} = \rho_2.$$

Now, we propose the following DC decomposition

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

with

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

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

*3.2.2. DCA for NLP2*

The DCA for NLP2 has the same framework as for NLP1 which yields the fixed point method as:

$$(x^{k+1}, y^{k+1}, w^{k+1}) \in argmin\{g(x, y, w) - \langle (x, y, w), \nabla h(x^k, y^k, w^k)\rangle : (x, y, w) \in C\}$$

with

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

Each iteration requires solving a strictly convex quadratic program over a polyhedral convex set $C$. The convergence theorem and the stopping criteria of DCA are the same as in the subsection 3.1.2.

### 3.3. *Initialization strategy*

We must find an initial point to start DCA. The choice of a "good" initial point is very important since it will affect the quality of the numerical solution, as well as the total number of iterations for the convergence of DCA. However, finding a good initial point for DCA is always an open question. For a general DC program, there is no deterministic method for finding a good initial point. Note that a good initial point means a point from which DCA can converge to a global optimal solution of the DC program.

In our specific problem, we propose three different estimation methods for finding an initial point.

Init1. Zero initial point.
Init2. Optimal solution of SDP:

$$\begin{aligned} (SDP) \qquad & z^* = \max z \\ & s.t.\ B - zD \succeq 0 \\ & \quad Bx - Ay \geq 0 \\ & \quad e^T x = 1 \\ & \quad x \geq 0, y \geq 0, u \geq z \geq l. \end{aligned} \tag{13}$$

Init3. We can also improve the initial point from the solution of (SDP) by solving the later convex QP:

$$(QP_{z^*}) x^* \in argmin\{x^T(B - z^*D)x : Bx - Ay \geq 0, e^T x = 1, (x, y) \geq 0\}.$$

The $(QP_{z^*})$ is a convex program since $B - z^*D \succeq 0$. It is easy to prove that if $(QP_{z^*})$ has zero optimal value, then its optimal solution $(x^*, y^* = z^*x^*, w^* = Bx^* - Ay^*, z*)$ should be also the optimal solution of NLP and $(x^*, \lambda^* = \frac{1}{z^*})$ is a solution of EiCP. Otherwise, we can use its solution as an initial point for starting DCA. In numerical computation, (SDP) can be solved by SeDuMi [19] and $(QP_{z^*})$ by CPLEX [3].

## 4. Numerical Simulations

In this section, we report some numerical simulation results. All experiments were realized on a PC equipped with Windows Vista OS, Intel Core2 Duo P8400 2.26 GHz, 4GB RAM memory. Our codes were implemented in MATLAB R2008a [4] (a C version is also developed). In the paper, we only give the results with Matlab version. The YALMIP optimization toolbox v3 [9] is used for modeling the mathematical programming problem, and for invoking CPLEX v12.1 [3] on Matlab to solve the convex QPs and LPs.

In our tests, $B$ is taken as a real positive definite matrix (often $I_n$) and $A$ is either an asymmetric matrix with randomly generated elements, or from the Matrix Market NEP repository in applied sciences and engineering [5].

### 4.1. *Tests of Asymmetric Matrices with Randomly Generated Uniformly Distributed Elements*

The first set of test problems uses $B = I_n$ and $A$ an asymmetric matrix whose elements were randomly generated and uniformly distributed in the interval $[0, 2]$. The order $n$ of the tested matrices is between 10 - 1000. In this set of tests, the existence of the solution of EiCP is guaranteed since $A \in C$ and $-A \in R_0$ , $B = I_n \in SC$ [2, 7]. Hence all NLP formulations proposed in our paper have zero global optimal values. For these rests, we fixed the parameters $\epsilon_1 = 10^{-6}, \epsilon_2 = 10^{-6}$ and $\epsilon_3 = 10^{-6}$ and used 0 as the initial point.

In Table 1, DCA shows a good performance for solving both the NLP1 and the NLP2 formulations. For this kind of random problems, DCA always converges to zero optimal value with few iterations (less than 10 iterations for NLP1 and 8 iterations for NLP2) and within very short computational time (less than 35.366 seconds for NLP1 and less than 31.895 seconds for NLP2). It seems that DCA for NLP2 often requires less number of iterations and it is a bit faster than DCA for NLP1. Moreover, when the order of the matrix becomes larger (more than 200), most of the cases only need one iteration. These remarkable results outperformed the Enumerative method, Branch-and-bound method, and BARON/GAMS presented in [7].

Table 1. Computational results of DCA for NLP1 and NLP2 with randomly generated uniformly distributed matrices and with 0 as starting point

| n | l | u | DCA for NLP1 | | | | DCA for NLP2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Iter | Obj | z* | T | Iter | Obj | z* | T |
| 10 | 7.51e-3 | 1.20e-2 | 10 | 1.27e-5 | 7.99e-3 | 0.661 | 8 | 1.16e-5 | 8.01e-3 | 0.465 |
| 30 | 1.18e-3 | 1.62e-3 | 3 | 1.09e-7 | 1.19e-3 | 0.220 | 2 | 1.22e-7 | 1.20e-3 | 0.127 |
| 50 | 4.00e-4 | 6.01e-4 | 2 | 1.63e-7 | 4.00e-4 | 0.169 | 2 | 1.69e-8 | 5.00e-4 | 0.149 |
| 70 | 2.00e-4 | 2.37e-4 | 2 | 2.98e-9 | 2.00e-4 | 0.216 | 2 | 2.87e-9 | 2.09e-4 | 0.190 |
| 90 | 1.19e-4 | 1.63e-4 | 2 | 3.33e-9 | 1.19e-4 | 0.229 | 2 | 3.35e-9 | 1.31e-4 | 0.196 |
| 100 | 9.77e-5 | 1.18e-4 | 2 | 6.35e-10 | 9.77e-5 | 0.244 | 2 | 8.69e-10 | 1.02e-4 | 0.227 |
| 200 | 2.48e-5 | 2.98e-5 | 1 | 1.11e-10 | 2.48e-5 | 0.481 | 1 | 1.65e-10 | 2.60e-5 | 0.615 |
| 300 | 1.11e-5 | 1.35e-5 | 1 | 2.39e-10 | 1.11e-5 | 0.985 | 1 | 2.09e-10 | 1.11e-5 | 0.966 |
| 400 | 6.26e-6 | 7.14e-6 | 1 | 4.69e-10 | 6.26e-6 | 1.934 | 1 | 7.04e-11 | 6.40e-6 | 2.522 |
| 500 | 4.01e-6 | 4.41e-6 | 1 | 1.39e-10 | 4.01e-6 | 3.736 | 1 | 3.66e-10 | 4.10e-6 | 4.488 |
| 600 | 2.78e-6 | 3.06e-6 | 1 | 4.80e-10 | 2.78e-6 | 11.684 | 1 | 6.77e-11 | 2.83e-6 | 10.369 |
| 700 | 2.04e-6 | 2.28e-6 | 1 | 4.31e-10 | 2.04e-6 | 12.003 | 1 | 7.20e-11 | 2.08e-6 | 10.101 |
| 800 | 1.56e-6 | 1.73e-6 | 1 | 7.32e-11 | 1.56e-6 | 18.875 | 1 | 1.53e-11 | 1.59e-6 | 17.699 |
| 900 | 1.23e-6 | 1.34e-6 | 1 | 2.56e-10 | 1.23e-6 | 31.316 | 1 | 1.04e-10 | 1.25e-6 | 30.813 |
| 1000 | 9.98e-7 | 1.09e-6 | 1 | 5.53e-10 | 9.98e-7 | 35.366 | 1 | 6.23e-11 | 1.01e-6 | 31.895 |

### 4.2. *Tests of Matrix Market NEP repository*

The second set of tests take $B = I_n$ and $A$ as an asymmetric matrix from the Matrix Market repository NEP (Non-Hermitian Eigenvalue Problem) collection (a testbed for the development of numerical algorithms for solving asymmetric eigenvalue problems and challenging non-hermitian eigenvalue problems in real applications). These matrices $A$ (have orders between $60 - 800$) come from various areas and are illustrated in Table 2:

Table 2. Matrices from Matrix Market NEP Collection

| Disciplines | Matrices |
|---|---|
| Electrical engineering | BFW62A, BFW398A, DWA512, DWB521, BFW782A |
| Computational fluid dynamics | TUB100, OLM100, RDB200, RDB450, BWM200, PDE225 |
| Computer science | LOP163 |
| Robotic control | RBS480A, RBS480B |
| Aeroelasticity | TOLS90, TOLS340 |
| Markov chain modeling | RW136 |

We test DCA for NLP1 with fixed parameters $\epsilon_1 = 10^{-5}, \epsilon_2 = 10^{-5}$ and $\epsilon_3 = 10^{-4}$. The numerical results are presented in Table 3. The column "T" in Table 3 represents the total computational time (seconds):

$$\text{T} = \text{Time for finding initial point} + \text{Time for solving NLP via DCA.}$$

Furthermore "It" is the number of iterations of DCA, "Obj" is the computed optimal value of $f$ obtained by DCA and "$z^*$" is the computed optimal value of $z$. We start DCA via three different initial point estimation methods Init$i, i = 1, 2, 3$ in subsection 3.3 and compare their numerical solutions. We observe that the different initial points yield different optimal solutions. We can summarize three major results as follows:

(1) In most of the problems, the method Init3 is less computational expensive and yields a better solution than the method Init2. The quality of the computed solution can be evaluated by its objective value. The smaller the value is, the better the solution will be.

(2) For the problems bwm200, rdb200, rdb400 and dwb512, the method Init3 gives directly a global optimal solution with zero optimal value, and DCA

Table 3. Numerical results for NEP repository with three initial point strategies

| Problem | size | Init1 (0-init) | | | | Init2 (SDP) | | | | Init3 (SDP+QP) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | T | It | Obj | $z^*$ | T | It | Obj | $z^*$ | T | It | Obj | $z^*$ |
| bfw62a | 62 | 25.02 | 328 | 2.40E-04 | 2.90E-01 | 8.08 | 100 | 2.10E-04 | 6.17E-01 | 15.46 | 193 | 2.12E-04 | 2.05E-01 |
| bfw62a | 62 | 25.02 | 328 | 2.40E-04 | 2.90E-01 | 8.08 | 100 | 2.10E-04 | 6.17E-01 | 15.46 | 193 | 2.12E-04 | 2.05E-01 |
| tols90 | 90 | 4.61 | 52 | 9.97E-05 | 6.59E-02 | 177.59 | 1989 | 6.38E-04 | 7.35E-02 | 94.38 | 1055 | 1.05E-03 | 2.21E-02 |
| olm100 | 100 | 12.79 | 146 | 2.04E-03 | 1.65E-03 | 24.93 | 283 | 1.25E-03 | 9.36E-02 | 12.92 | 145 | 2.02E-03 | 3.99E-03 |
| tub100 | 100 | 0.35 | 4 | 9.80E-03 | 7.90E-08 | 16.35 | 182 | 1.98E-03 | 9.82E-02 | 4.83 | 54 | 5.24E-04 | 9.40E-02 |
| rw136 | 136 | 3.20 | 32 | 5.46E-03 | 3.98E-04 | 4.18 | 41 | 1.81E-04 | 9.44E-01 | 0.85 | 8 | 8.97E-05 | 9.43E-01 |
| lop163 | 163 | 3.28 | 30 | 4.66E-03 | 2.99E-04 | 1.91 | 17 | 8.92E-05 | 9.02E-01 | 9.54 | 84 | 2.56E-03 | 9.00E-01 |
| bwm200 | 200 | 0.26 | 2 | 5.00E-03 | 3.57E-09 | 9.14 | 74 | 3.22E-04 | 2.37E-01 | 0.15 | 1 | 1.13E-07 | 2.37E-01 |
| rdb200 | 200 | 0.25 | 1 | 5.00E-03 | 2.15E-09 | 10.50 | 83 | 2.32E-04 | 2.38E-01 | 0.14 | 1 | 1.22E-07 | 2.38E-01 |
| pde225 | 225 | 2.58 | 19 | 4.01E-03 | 3.37E-05 | 17.65 | 92 | 4.05E-03 | 1.09E-01 | 2.88 | 21 | 3.53E-03 | 1.05E-01 |
| tols340 | 340 | 3.29 | 18 | 2.38E-03 | 1.17E-05 | 204.82 | 939 | 6.57E-03 | 4.35E-03 | 36.82 | 18 | 2.37E-03 | 2.33E-05 |
| bfw398a | 398 | 1.09 | 5 | 2.45E-03 | 7.09E-07 | 100.67 | 305 | 1.49E-03 | 2.24E-01 | 37.12 | 11 | 1.74E-03 | 9.61E-02 |
| rdb450 | 450 | 0.49 | 2 | 2.22E-03 | 1.98E-09 | 68.15 | 63 | 1.42E-04 | 2.38E-01 | 53.77 | 1 | 6.62E-08 | 2.38E-01 |
| rbs480a | 480 | 2.55 | 7 | 4.89E-04 | 9.86E-07 | 898.14 | 2500 | 2.39E-02 | 1.49E-01 | 70.33 | 8 | 3.27E-04 | 1.85E-03 |
| rbs480b | 480 | 2.10 | 6 | 1.71E-03 | 7.89E-08 | 231.07 | 475 | 1.73E-02 | 8.60E-03 | 86.94 | 47 | 7.10E-04 | 1.79E-03 |
| dwa512 | 512 | 0.54 | 2 | 1.95E-03 | 5.13E-08 | 87.28 | 64 | 1.79E-03 | 1.30E+00 | 75.29 | 3 | 1.04E-04 | 1.29E+00 |
| dwb512 | 512 | 3.47 | 13 | 1.72E-03 | 1.67E-05 | 80.32 | 18 | 3.34E-04 | 1.02E+00 | 76.12 | 1 | 1.33E-08 | 1.02E+00 |
| bfw782a | 782 | 0.93 | 2 | 1.26E-03 | 2.58E-08 | 378.51 | 229 | 1.03E-03 | 2.37E-01 | 250.27 | 4 | 1.11E-03 | 8.37E-02 |

required only one iteration. It means that the SDP+QP method could give the best initial point.

(3) For problems of large size (larger than 200), the method Init1 is much less computational expensive (within $0.2 - 3.5$ seconds and $1 - 20$ iterations) than the other two methods. However, the last two methods often get better solutions than Init1.

It should be noted that solving a SDP is computational expensive, especially for large-scale problems. Most of the computational time for methods Init2 and Init3 is spent on solving SDP for searching an initial point. For instance, for the problem "bfw782a", the method with Init3 spent 248.348 seconds for solving SDP and only 1.918 seconds for DCA. The main procedure of DCA is less computational expensive and can handle very well large-scale nonlinear programs. The open question is how to find a less expensive "good" initial point estimation method for starting DCA. Despite the local optimality of DCA, we observe in the tables 1 and 3 that if the solution of EiCP exists, DCA can almost always converge to the global optimal solution of NLP with our proposed starting point estimation methods since the objective values are close to zero (less than $10^{-3}$). Although solving SDP is computational expensive for large-scale problems, it is a good method for initial point estimation.

## 5. Conclusion and Perspective

In this paper, we introduced three nonlinear programming (NLP) formulations for the asymmetric EiCP, and reformulated these nonlinear programs as DC programs that can be solved by efficient DC Algorithms (DCA). Three initialization strategies of DCA are investigated. Some numerical simulations for randomly generated problems and real world problems show a good performance of our methods. DCA almost always gets a global optimal solutions with zero optimal value of NLPs (solution of EiCP) within a short computational time and is especially efficient for relatively large-scale problems.

The third NLP formulation is proposed in this paper but the use of DCA for solving the corresponding DC program has not been studied. The gradient and the Hessian matrix of $f$ can be computed in a similar way as in NLP2. However, there is difficulty for using this formulation which lies on how to handle efficiently this type of nonconvex constraint in the DC program. This topic will be investigated in the future.

## References


[1] S. Adly, A. Seeger. *A nonsmooth algorithm for cone-constrained eigenvalue problems*, to appear in Computational Optimization and Applications.

[2] R.W. Cottle, J.S. Pang and R.E. Stone. *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[3] CPLEX 12.1, *ILOG CPLEX Documentation*, ILOG IBM., 2009 `http://www-01.ibm.com/software/integration/optimization/cplex/`

[4] MATLAB R2008a, *Documentation and User Guides.* `http://www.mathworks.com/`

[5] Matrix Market - NEP Collection, available at `http://math.nist.gov/MatrixMarket/data/NEP/`

[6] J.J Judice, H.D. Sherali, and I.M. Ribeiro. *The eigenvalue complementarity problem*, Computational Optimization and Applications. 37, pp. 139-156, 2007.

[7] J.J Judice, H.D. Sherali, I.M. Ribeiro, and S.S. Rosa. *On the asymmetric eigenvalue complementarity problem*, Optimization Methods & Software. 24, pp. 549-568, 2009.

[8] H.A. Le Thi, M. Moeini, T. Pham Dinh, and J.J. Judice. *A DC programming approach for solving the symmetric Eigenvalue Complementarity Problem*, Computational Optimization and Applications. Published online: 22 December 2010.

[9] J. Löfberg. *YALMIP: A Toolbox for Modeling and Optimization in MATLAB*, In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004. Software available at `http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php`

[10] Y.S. Niu. *Programmation DC & DCA en Optimisation Combinatoire et Optimisation Polynomiale via les Techniques de SDP*, PhD thesis, INSA de Rouen, France, 2010.

[11] T. Pham Dinh, H.A. Le Thi. *Convex analysis approach to DC programming: Theory, Algorithms and Applications*, Acta Mathematica Vietnamica. 22(1), pp. 289-355, 1997.

[12] T. Pham Dinh, H.A. Le Thi. *DC optimization algorithms for solving the trust region subproblem*, SIAM Journal on Optimization. 8, pp. 476-507, 1998.

[13] T. Pham Dinh, H.A. Le Thi. *DC Programming. Theory, Algorithms, Applications: The State of the Art*, First International Workshop on Global Constrained Optimization and Constraint Satisfaction, Nice, October 2-4, 2002.

[14] T. Pham Dinh, H.A. Le Thi. *The DC programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems*, Annals of Operations Research. 133, pp. 23-46, 2005.

[15] T. Pham Dinh, Y.S. Niu. *An efficient DC programming approach for portfolio decision with higher moments*, Computational Optimization and Applications. Published online: 23 December 2010.

[16] A. Pinto de Costa, I. Figueiredo, J.J. Judice and J.C. Marins. *The directional instability problem in systems with frictional contacts*, Computer Methods in Applied Mechanics and Engineering. 193, pp. 357-384, 2004.

[17] A. Pinto de Costa, A. Seeger. *Cone-constrained eigenvalue problems: Theory and algorithms*, Computational Optimization and Applications. 45, pp. 25-57, 2010.

[18] M. Queiroz, J.J Judice, and J.C. Humes. *The symmetric eigenvalue complementarity problem*, Mathematics of Computation. 73, pp. 1849-1863, 2003.

[19] SeDuMi 1.3, `http://sedumi.ie.lehigh.edu/`

[20] A. Seeger. *Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions*, Linear Algebra and its Applications. 292, pp. 1-14, 1999.

[21] A. Seeger, M. Torki. *On eigenvalues induced by a cone constraint*, Linear Algebra and its Applications. 372, pp. 181-206, 2003.

[22] A. Seeger, M. Torki. *Local minima of quadratic forms on convex cones*, Journal of Global Optimization. 44, pp. 1-28, 2009.

[23] Y. Zhou, M.S. Gowda. *On the finiteness of the cone spectrum of certain linear transformations on Euclidean Jordan algebras*, Linear algebra and its Applications. 431, pp. 772-782, 2009.