Numerical Algorithms

On the Numerical Solution of the Quadratic Eigenvalue Complementarity Problem

--Manuscript Draft--

Manuscript Number:	
Full Title:	On the Numerical Solution of the Quadratic Eigenvalue Complementarity Problem
Article Type:	Original Research
Keywords:	Eigenvalue Problems; Complementarity Problems; Nonlinear Programming; Global Optimization.
Corresponding Author:	Valentina Sessa, Ph.D. IMPA Rio de Janeiro, RJ BRAZIL
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	IMPA
Corresponding Author's Secondary Institution:	
First Author:	Alfredo Noel Iusem
First Author Secondary Information:	
Order of Authors:	Alfredo Noel Iusem
	Joaquim João Júdice
	Valentina Sessa, Ph.D.
	Hanif D. Sherali
Order of Authors Secondary Information:	
Abstract:	The Quadratic Eigenvalue Complementarity Problem (QEiCP) is an extension of the Eigenvalue Complementarity Problem (EiCP) that has been introduced recently. Similar to the EiCP, the QEiCP always has a solution under reasonable hypotheses on the matrices included in its definition. This has been established in a previous paper by reducing a QEiCP of dimension \$n\$ to a special \$2n\$-order EiCP. In this paper we propose an enumerative algorithm for solving the QEiCP by exploiting this equivalence with an EiCP. The algorithm seeks a global minimum of a special Nonlinear Programming Problem (NLP) with a known global optimal value. The algorithm is shown to perform very well in practice but in some cases terminates with only an approximate optimal solution to NLP. Hence, we propose a hybrid method that combines the enumerative method with a fast and local semi-smooth method to overcome the latter drawback. This algorithm is also shown to be useful for computing a positive eigenvalue for an EiCP under similar assumptions. Computational experience is reported to demonstrate the efficacy and efficiency of the hybrid enumerative method for solving the QEiCP.

Numerical Algorithms manuscript No.
(will be inserted by the editor)

On the Numerical Solution of the Quadratic Eigenvalue Complementarity Problem

Alfredo N. Iusem · **Joaquim J. Júdice** ·
Valentina Sessa · **Hanif D. Sherali**

Received: date / Accepted: date

Abstract The Quadratic Eigenvalue Complementarity Problem (QEiCP) is an extension of the Eigenvalue Complementarity Problem (EiCP) that has been introduced recently. Similar to the EiCP, the QEiCP always has a solution under reasonable hypotheses on the matrices included in its definition. This has been established in a previous paper by reducing a QEiCP of dimension n to a special $2n$-order EiCP. In this paper we propose an enumerative algorithm for solving the QEiCP by exploiting this equivalence with an EiCP. The algorithm seeks a global minimum of a special Nonlinear Programming Problem (NLP) with a known global optimal value. The algorithm is shown to perform very well in practice but in some cases terminates with only an approximate optimal solution to NLP. Hence, we propose a hybrid method that combines the enumerative method with a fast and local semi-smooth method to overcome the latter drawback. This algorithm is also shown to be useful for computing a positive eigenvalue for an EiCP under similar assumptions. Computational experience is reported to demonstrate the efficacy and efficiency of the hybrid enumerative method for solving the QEiCP.

Keywords Eigenvalue Problems · Complementarity Problems · Nonlinear Programming · Global Optimization.

Mathematics Subject Classification (2000) 90B60 · 90C33 · 90C30 · 90C26

A. N. Iusem
Instituto de Matématica Pura e Aplicada (IMPA), Estrada Dona Castorina 110, Rio de Janeiro, RJ, CEP 22460-320, Brazil, E-mail: iusp@impa.br. The work of this author was partially supported by CNPq grant no. 301280/86.,

J. J. Júdice
Instituto de Telecomunicações, Portugal, E-mail: joaquim.judice@co.it.pt.

V. Sessa
Instituto de Matématica Pura e Aplicada (IMPA), Estrada Dona Castorina 110, Rio de Janeiro, RJ, CEP 22460-320, Brazil, E-mail: valsessa@impa.br.

H. D. Sherali
Grado Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA, USA E-mail: hanifs@vt.edu.

1 Introduction

Given matrices B, $C \in \mathbb{R}^{n\times n}$, the *Eigenvalue Complementarity Problem* (denoted EiCP(B,C); see, e.g., [21] and [22]), consists of finding $(\lambda,x,w) \in \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^n$ such that

$$w = \lambda Bx - Cx \tag{1}$$
$$w \geq 0, x \geq 0 \tag{2}$$
$$x^\top w = 0 \tag{3}$$
$$e^\top x = 1, \tag{4}$$

with $e = (1,1,\ldots,1)^\top \in \mathbb{R}^n$, where constraint (4) is introduced, without loss of generality, to prevent the x-component of a solution to vanish. Usually, the matrix B is assumed to be positive definite (PD). This problem has many applications in engineering (see [19], [22]). If a triplet (λ,x,w) solves EiCP, then the scalar λ is called a *complementary eigenvalue* and x is a *complementary eigenvector* associated with λ for the pair (B,C). The condition $x^\top w = 0$ and the nonnegative requirements on x and w imply that either $x_i = 0$ or $w_i = 0$ for $1 \leq i \leq n$. These pairs of variables are called complementary. The EiCP always has a solution provided that the matrix B is PD [13].

If the matrices B and C are both symmetric, then EiCP is called symmetric and reduces to the problem of finding a *stationary point* (SP) of the so-called Rayleigh Quotient function on the simplex Ω (see, e.g. [21], [22]), which is essentially a SP of the following standard quadratic fractional program:

$$\begin{aligned} \text{Maximize } & \frac{x^\top Cx}{x^\top Bx} \\ \text{subject to } & e^\top x = 1 \\ & x \geq 0. \end{aligned} \tag{5}$$

A number of techniques have been proposed for solving EiCP and its extensions; see, e.g., [1], [2], [9], [10], [11], [12], [13], [14], [18], [20], and [24]. As expected, the symmetric EiCP is easier to solve.

Recently an extension of the EiCP has been introduced in [23], where some applications are highlighted, which is called the *Quadratic Eigenvalue Complementarity Problem* (QEiCP). This problem differs from the EiCP through the existence of an additional quadratic term in λ. Its formal definition follows.

Given $A,B,C \in \mathbb{R}^{n\times n}$, QEiCP($A,B,C$) consists of finding $(\lambda,x,w) \in \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^n$ such that

$$w = \lambda^2 Ax + \lambda Bx + Cx, \tag{6}$$
$$w \geq 0, x \geq 0, \tag{7}$$
$$x^\top w = 0, \tag{8}$$
$$e^\top x = 1, \tag{9}$$

where, as before, $e = (1,1,\ldots,1)^\top \in \mathbb{R}^n$. Note that QEiCP($A,B,C$) reduces to EiCP($B,-C$) when $A = 0$. Furthermore, finding a positive complementary eigenvalue for EiCP(B,C) is equivalent to computing a nonzero quadratic complementary eigenvalue of QEiCP($B,0,-C$). The λ-component of a solution to QEiCP(A,B,C) is called a *quadratic complementary eigenvalue* for A,B,C, and the x-component is called a *quadratic complementary eigenvector* for A,B,C associated with λ.

The case of the symmetric QEiCP, i.e., when A,B, and C are symmetric matrices, and $C = -I$, where I is the identity matrix, has been analyzed in [8], where each instance of QEiCP with $n \times n$ matrices is related to an instance of EiCP with $2n \times 2n$ matrices. In this paper, we remove the symmetry assumption, and focus on the general QEiCP. In [3], a relation between an n-dimensional QEiCP and certain $2n$-dimensional instances of EiCP was introduced. This "reduction" of QEiCP to EiCP was suggested mainly with a theoretical purpose in mind, namely, to establish necessary and/or sufficient conditions on A,B,C that ensure the existence of solutions to QEiCP(A,B,C). In particular, QEiCP has positive and negative quadratic complementary eigenvalues if $A \in$ PD and C is not an S_0-matrix, i.e., there exists no $0 \neq x \geq 0$ such that $Cx \geq 0$ [3]. Note that these considerations should be considered as an extension of the sufficient conditions for the symmetric QEiCP, as $C = -I$ is not an S_0 matrix. Furthermore, these conditions imply that an asymmetric EiCP(B,C) has at least a positive complementary eigenvalue if $B \in$ PD and $C^\top$ is an S-matrix, i.e., there exists a $x \geq 0$ such that $C^\top x > 0$. This result is proved later in this paper along with a discussion on its importance in practice. Recall that some applications of the EiCP require the complementary eigenvalue to be positive [19].

Another set of sufficient conditions for the existence of solutions to QEiCP, called *co-regularity* and *co-hyperbolicity*, was proposed in [23]. An enumerative method and a hybrid algorithm for QEiCP, combining this enumerative method with a semi-smooth approach, have been introduced in [9] and [10]. These methods are able to solve the QEiCP when the co-regularity and co-hyperbolicity conditions are assumed to hold. In [3], the numerical solution of QEiCP by solving its equivalent $2n$-dimensional EiCP referred to above has been discussed. Variational Inequality (VI) and Nonlinear Programming (NLP) formulations have been introduced for this purpose. Numerical experiments reported in [3] clearly indicate that the NLP formulation seems to be more effective, particularly since the global optimal value is known to be zero. In this paper, we propose an enumerative method for finding a global minimum of such an NLP that exploits this desirable feature of NLP. This algorithm is based on ideas similar to the ones discussed in [9] and it computes stationary points of the objective function of NLP until it finds one that achieves the known zero optimal value. As in [10], this method can be combined with the semi-smooth method similar to the one introduced in [23] in order to enhance its computational efficiency. Numerical results included in the paper indicate the efficacy and efficiency of the hybrid (enumerative plus semi-smooth) method for the solution of the QEiCP when $A \in$ PD and C is not an S_0-matrix.

The organization of the remainder of this paper is as follows. In Section 2, the $2n$-dimensional EiCPs that are equivalent to the QEiCP and their NLP formulations are introduced. The enumerative method is described in Sections 3 and 4. The semi-smooth algorithm for the $2n$-dimensional EiCPs is introduced in Section 5. The hybrid approach combining the enumerative and the semi-smooth methods is discussed in Section 6. The computation of a positive complementary eigenvalue for an EiCP is discussed in Section 7. Numerical results are reported in Section 8, and some concluding comments are given in Section 9.

2 A Nonlinear Programming Formulation

Consider QEiCP(A,B,C) with $A,B,C \in \mathbb{R}^{n\times n}$ and assume that A is a PD matrix and C is not an S_0-matrix, that is

(i) $x^\top Ax > 0$ for all $x \neq 0$
(ii) there is no $0 \neq x \geq 0$ such that $Cx \geq 0$.

Note that it is relatively easy to verify whether a given matrix is PD or S_0. The $LDL^\top$ decomposition of the symmetric form of A is required for checking the first property while the solution of a linear program suffices for checking the second property.

As in [3], we introduce the $2n$-dimensional EiCP(D,G) and EiCP(D,H) formulation, where

$$D = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}, \quad G = \begin{bmatrix} -B & -C \\ I & 0 \end{bmatrix}, \quad H = \begin{bmatrix} B & -C \\ I & 0 \end{bmatrix}, \tag{10}$$

with I being the identity matrix of order n. Note that the matrix D of the λ-term of the two EiCPs is PD. This means that these EiCPs have at least one solution [13].

In order to see the implementation of solving QEiCP by finding a solution to these EiCPs, we write the EiCP(D,G) as follows:

$$\lambda \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} -B & -C \\ I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} w \\ t \end{bmatrix} \tag{11a}$$

$$e^\top y + e^\top x = 1 \tag{11b}$$

$$y^\top w = x^\top t = 0 \tag{11c}$$

$$x, y, w, t, \lambda \geq 0. \tag{11d}$$

Then the following result holds [3]:

Theorem 1 *Let $A \in PD$ and $C \notin S_0$. If $(\bar{\lambda}, \bar{x}, \bar{y})$ is a solution of EiCP(D,G) then:*

(i) $\bar{\lambda} > 0$ and $\bar{y} = \bar{\lambda}\bar{x}$.
(ii) $\bar{\lambda}$ is a quadratic complementary eigenvalue of QEiCP and $(1+\bar{\lambda})\bar{x}$ is an associated eigenvector.

Note that a similar result holds for EiCP(D,H) with $-\bar{\lambda}$ instead of $\bar{\lambda}$ in (ii) (however, the eigenvector has the same form). Therefore, if $A \in$ PD and $C \notin S_0$, the QEiCP has at least a positive and a negative quadratic complementary eigenvalue, which can be computed by solving EiCP(D,G) and EiCP(D,H), respectively. In this paper, we concentrate our attention solely on the computation of a positive quadratic complementary eigenvalue as the case of a negative eigenvalue is similar.

Consider again the EiCP (11). By Theorem 1, $t = 0$ in any solution of the EiCP. If we introduce the vector v such that $v = \lambda y$, then we get the following Nonlinear Programmming Formulation of the EiCP (11) introduced in [3]:

$$\begin{aligned}
\mathbf{NLP}_1 : \text{Minimize} \quad & f(x,y,v,w,\lambda) = (y-\lambda x)^\top (y-\lambda x) + (v-\lambda y)^\top (v-\lambda y) \\
& \qquad + (x+y+v)^\top w & \text{(12a)} \\
\text{subject to} \quad & w = Av + By + Cx & \text{(12b)} \\
& e^\top y + e^\top x = 1 & \text{(12c)} \\
& e^\top v + e^\top y = \lambda & \text{(12d)} \\
& x,y,v,w \geq 0. & \text{(12e)}
\end{aligned}$$

Furthermore, the following result holds [3]:

Theorem 2 *Let A be strictly copositive and $C \notin S_0$. Then the nonlinear problem $\mathbf{NLP}_1$ in (12) has a global minimum and $(\bar{\lambda}, (1+\bar{\lambda})\bar{x})$ is a solution of QEiCP.*

Proof See Proposition 7 in [3]. □

In the next two sections, we introduce an enumerative method for finding a global minimum for $\mathbf{NLP}_1$. Since the global optimal value of $\mathbf{NLP}_1$ is equal to zero, the algorithm computes stationary points for $\mathbf{NLP}_1$ in a systematic way until finding one with a null objective function value (or a value smaller than a prescribed tolerance). These stationary points are associated with the nodes of a binary tree that is generated according to the branching strategy defined in [13]. Bounds on the complementary eigenvalue are required in order to generate constraints based on the Reformulation-Linearization Technique (RLT) [25] that facilitate the search for a global minimum of $\mathbf{NLP}_1$. In the next section, we discuss how these bounds and RLT constraints are generated. The enumerative algorithm is then described in Section 4.

3 Lower and upper bounds for a quadratic complementary eigenvalue

3.1 Computing an upper bound

The next theorem provides an upper bound u for a quadratic complementary eigenvalue λ.

Theorem 3 *Let $p_i = 1 + \sum_{j=1}^{n} (\max\{0, -b_{ij}\} + \max\{0, -c_{ij}\})$ for all $i = 1, \ldots, n$, and let $p \in \mathbb{R}^n$ be a vector with components p_i. Then we can take*

$$u = \frac{p^\top \bar{y}}{\bar{y}^\top A \bar{y} + \bar{x}^\top \bar{x}}, \tag{13}$$

where $(\bar{x}, \bar{y})$ is a stationary point of the following nonlinear problem:

$$\begin{aligned}
\mathbf{NLP}_2 : \textit{Maximize} \quad & \frac{p^\top y}{y^\top A y + x^\top x} \\
\textit{subject to} \quad & e^\top y + e^\top x = 1 \\
& x, y \geq 0.
\end{aligned}$$

Proof If λ is a solution of EiCP(D,G), given by (11), then

$$\exists z \in \Delta : \quad \lambda = \frac{z^\top Gz}{z^\top Dz}, \tag{14}$$

with $\Delta = \{z \in \mathbb{R}^{2n} : e^\top z = 1, z \geq 0\}$, $z = (x,y)$, and with G and D given by (10). Hence,

$$z^\top Gz = -y^\top By - y^\top Cx + x^\top y = y^\top(-By - Cx + x) \tag{15a}$$

$$z^\top Dz = y^\top Ay + x^\top x. \tag{15b}$$

But

$$\begin{aligned}(-By - Cx + x)_i &= \sum_{j=1}^{n}(-b_{ij}y_j - c_{ij}x_j) + x_i \\ &\leq \sum_{j=1}^{n} \max\{0, -b_{ij}\}y_j + \max\{0, -c_{ij}\}x_j + x_i \\ &\leq p_i, \quad \forall i = 1,\ldots,n,\end{aligned}$$

where p_i $(p_i,\ i = 1,\ldots,n)$ is defined in the theorem. Since $0 \leq y_i \leq 1$ and $0 \leq x_i \leq 1$ for all $i = 1,\ldots,n$, then $z^\top Gz \leq p^\top y$. Now, consider the function

$$f(x,y) = \frac{p^\top y}{y^\top Ay + x^\top x}. \tag{16}$$

Since A is positive definite then the expression in the denominator of (16) is strictly convex on the simplex Δ. Hence f is pseudo-concave, [1], and any stationary point $(\bar{x},\bar{y})$ of f in Δ is a global maximum. Thus, an upper bound can be computed as in (13). □

3.2 Computing a lower bound

For the computation of the lower bound l, consider the following linear program:

$$\begin{aligned}\mathbf{LP}: \quad &\text{Minimize} \quad && e^\top v + e^\top y \\ &\text{subject to} && Av + By + Cx \geq 0 \\ &&& e^\top y + e^\top x = 1 \\ &&& x, y, v, w \geq 0.\end{aligned}$$

An optimal solution to **LP** provides a lower bound l on λ, as established by the following results:

Theorem 4 *If A is PD,* ***LP*** *has an optimal solution.*

Proof Let $\bar{x},\bar{y} \geq 0$ such that $e^T\bar{x}+e^T\bar{y}=1$. Since A is PD, the system

$$\begin{aligned} Av+(B\bar{y}+C\bar{x}) &\geq 0 \\ v &\geq 0 \end{aligned}$$

has a solution [6]. Hence **LP** has an optimal solution, since it is feasible and the objective function is bounded from below on its feasible set. □

Theorem 5 *If $C \notin S_0$, then **LP** has a positive optimal value.*

Proof **LP** has a zero optimal value if and only if $y=v=0$ in any optimal solution. In this case, there must exist an $x \geq 0$ such that $Cx \geq 0$ and $e^\top x = 1$. This is impossible because $C \notin S_0$. □

Thus, the lower bound l, defined by the optimal value of **LP**, exists and is strictly positive when $A \in$ PD and $C \notin S_0$.

3.3 Reformulation-Linearization Technique (RLT) constraints

Based on the lower and the upper bounds on λ derived above, an additional constraint $l \leq \lambda \leq u$ can be added to the nonlinear problem $\mathbf{NLP}_1$. Furthermore, since $y=\lambda x$ and $v=\lambda y$, the following RLT bound-factor constraints [25] can also be added

$$lx_i \leq y_i \leq ux_i \tag{17a}$$

$$ly_i \leq v_i \leq uy_i \tag{17b}$$

$$l(1-x_i) \leq (\lambda - y_i) \leq u(1-x_i) \tag{17c}$$

$$l(1-y_i) \leq (\lambda - v_i) \leq u(1-y_i) \tag{17d}$$

for each $i=1,\ldots,n$.

By incorporating these constraints, we obtain the following augmented nonlinear program:

$$\begin{aligned}
\mathbf{NLP}_3: \text{Minimize} \quad & f(x,y,v,w,\lambda) = (y-\lambda x)^\top (y-\lambda x) + (v-\lambda y)^\top (v-\lambda y) \\
& \qquad + (x+y+v)^\top w && \text{(18a)} \\
\text{subject to} \quad & w = Av+By+Cx && \text{(18b)} \\
& e^\top y + e^\top x = 1 && \text{(18c)} \\
& e^\top v + e^\top y = \lambda && \text{(18d)} \\
& l \leq \lambda \leq u && \text{(18e)} \\
& lx_i \leq y_i \leq ux_i, \quad \forall\, i=1,\ldots,n && \text{(18f)} \\
& ly_i \leq v_i \leq uy_i, \quad \forall\, i=1,\ldots,n && \text{(18g)} \\
& l(1-x_i) \leq (\lambda - y_i) \leq u(1-x_i), \quad \forall\, i=1,\ldots,n && \text{(18h)} \\
& l(1-y_i) \leq (\lambda - v_i) \leq u(1-y_i), \quad \forall\, i=1,\ldots,n && \text{(18i)} \\
& (x,y,v,w) \geq 0. && \text{(18j)}
\end{aligned}$$

4 An enumerative algorithm for QEiCP

In this section, we introduce an enumerative algorithm for solving the nonlinear problem $\mathbf{NLP}_3$, which explores a binary tree that is constructed under two jointly managed branching strategies. The first is based on the complementarity conditions between the variables w and x, i.e., either $w_i = 0$ or $x_i = y_i = v_i = 0$ for each $i = 1, \ldots, n$ as $y_i = \lambda x_i$ and $v_i = \lambda y_i$ for each $i = 1, \ldots, n$. The second branching strategy consists of partitioning the interval $[l, u]$ for λ. This algorithm is based on ideas similar to the enumerative algorithm of EiCP proposed in [13].

Define the sets I and J that record the w_i- and (x_i, y_i, v_i)-variables that are currently set to zero, respectively. At each node of the tree we examine NLP_3 with λ constrained in the interval $[\bar{l}, \bar{u}] \subseteq [l, u]$ along with the following constraints:

$$\begin{aligned}
\bar{l}x_i \le y_i \le \bar{u}x_i, &\qquad \forall\, i \in \bar{J} \\
\bar{l}y_i \le v_i \le \bar{u}y_i, &\qquad \forall\, i \in \bar{J} \\
\bar{l}(1 - x_i) \le (\lambda - y_i) \le \bar{u}(1 - x_i), &\qquad \forall\, i \in \bar{J} \\
\bar{l}(1 - y_i) \le (\lambda - v_i) \le \bar{u}(1 - y_i), &\qquad \forall\, i \in \bar{J} \\
v_i = y_i = x_i = 0, &\quad \forall i \in J \\
w_i = 0, &\quad \forall i \in I,
\end{aligned}$$

where $l \le \bar{l} \le \bar{u} \le u$, $I \subseteq \{1, \ldots, n\}$, $J \subseteq \{1, \ldots, n\}$, $\bar{J} = \{1, \ldots, n\} \setminus J$ and $I \cap J = \emptyset$. Consider also the sets $K = I \cup J$, $\bar{K} = \{1, \ldots, n\} \setminus K$ and $\bar{I} = \{1, \ldots, n\} \setminus I$. Then, at each node k of the binary tree, we examine the following nonlinear problem:

$$\begin{aligned}
\mathbf{NLP_4(k)}: \text{Minimize} \quad & f(x, y, v, w, \lambda) = (y - \lambda x)^\top (y - \lambda x) + (v - \lambda y)^\top (v - \lambda y) \\
& \qquad + (x + y + v)^\top w && \text{(19a)} \\
\text{subject to} \quad & w = Av + By + Cx && \text{(19b)} \\
& e^\top y + e^\top x = 1 && \text{(19c)} \\
& e^\top v + e^\top y = \lambda && \text{(19d)} \\
& \bar{l} \le \lambda \le \bar{u} && \text{(19e)} \\
& \bar{l}x_i \le y_i \le \bar{u}x_i, \qquad \forall\, i \in \bar{J} && \text{(19f)} \\
& \bar{l}y_i \le v_i \le \bar{u}y_i, \qquad \forall\, i \in \bar{J} && \text{(19g)} \\
& \bar{l}(1 - x_i) \le (\lambda - y_i) \le \bar{u}(1 - x_i), \qquad \forall\, i \in \bar{J} && \text{(19h)} \\
& \bar{l}(1 - y_i) \le (\lambda - v_i) \le \bar{u}(1 - y_i), \qquad \forall\, i \in \bar{J} && \text{(19i)} \\
& (x, y, v, w) \ge 0 && \text{(19j)} \\
& v_i = y_i = x_i = 0, \quad \forall j \in J && \text{(19k)} \\
& w_i = 0, \quad \forall i \in I. && \text{(19l)}
\end{aligned}$$

The steps of the algorithm are as follows:

Algorithm 1 Enumerative algorithm

▷ **Step** 0 **(Initialization)**

Set ε_1 and ε_2 such that $0 < \varepsilon_1 < \varepsilon_2$.
Set $k = 1$, $I = \emptyset$, $J = \emptyset$ and find a stationary point $(\bar{x}, \bar{y}, \bar{v}, \bar{w}, \bar{\lambda})$ of $\mathbf{NLP}_4(1)$.
if $\mathbf{NLP}_4(1)$ is infeasible **then**
  QEiCP has no solution; terminate.
else
  Let $L = \{1\}$ be the set of open nodes.
  Let $UB(1) = f(\bar{x}, \bar{y}, \bar{v}, \bar{w}, \bar{\lambda})$.
  Let $N = 1$ be the number of generated nodes.
end if
Let

$$\theta_1 = \max\{\bar{w}_i \bar{x}_i : i \in \bar{K}\} = \bar{w}_r \bar{x}_r \tag{20}$$
$$\theta_2 = \max\{|\bar{v}_i - \bar{\lambda}\bar{y}_i|, |\bar{y}_i - \bar{\lambda}\bar{x}_i| : i \in \bar{J}\} \tag{21}$$

while $(\theta_1 > \varepsilon_1$ OR $\theta_2 > \varepsilon_2)$ **do**

▷ **Step** 1 **(Choice of node)**

  if $L = \emptyset$ **then**
    terminate; QEiCP has no solution.
  else
    Select $k \in L$ such that $UB(k) = \min\{UB(i) : i \in L\}$.
    Let $(\bar{x}, \bar{y}, \bar{v}, \bar{w}, \bar{\lambda})$ be the stationary point that was previously found at this node.
    If $k \neq 1$, compute θ_1 and θ_2 in (20) and (21), respectively.
  end if

▷ **Step** 2**: (Branching rule)**

  if $(\theta_1 > \theta_2)$ **then**
    Branch on the complementarity variables $\bar{w}_r$ and $(\bar{x}_r, \bar{y}_r, \bar{v}_r)$ associated with θ_1 and generate two new nodes $N+1$ and $N+2$.
  else if $(\theta_1 \leq \theta_2)$ **then**
    Partition the interval $[\bar{l}, \bar{u}]$ for $\bar{\lambda}$ at node k into $[\bar{l}, \tilde{\bar{\lambda}}]$ and $[\tilde{\bar{\lambda}}, \bar{u}]$ to generate
    two new nodes $N+1$ and $N+2$, where

$$\tilde{\lambda} = \begin{cases} \bar{\lambda} & \text{if } \min\{(\bar{\lambda} - \bar{l}), (\bar{u} - \bar{\lambda})\} \geq 0.1(\bar{u} - \bar{l}) \\ \frac{\bar{u}+\bar{l}}{2} & \text{otherwise.} \end{cases}$$

  end if

▷ **Step** 3 **(Solve, Update and Queue)**

  For each of $p = N+1$ and $p = N+2$, find a stationary point $(\tilde{x}, \tilde{y}, \tilde{v}, \tilde{w}, \tilde{\bar{\lambda}})$ of $\mathbf{NLP}_4$(p).
  If $\mathbf{NLP}_4$(p) is feasible, set $L = L \cup \{p\}$ and $UB(p) = f(\tilde{x}, \tilde{y}, \tilde{v}, \tilde{w}, \tilde{\bar{\lambda}})$.
  Set $L = L \setminus \{k\}$.
end while

We remark that if the algorithm terminates successfully, then $\bar{\lambda}$ is a quadratic complementarity eigenvalue for (A,B,C) (within the tolerance ε_2) and $(1+\bar{\lambda})\bar{x}$ is the corresponding quadratic complementarity eigenvector. The convergence of Algorithm 1 follows from Theorem 4.1 in [13].

Another strategy for selecting the branching decision at in each iteration could be to compare θ_1 with $\varepsilon_1/\varepsilon_2\ \theta_2$ instead of comparing it directly with θ_2. Such a scaling strategy could help make the comparison between θ_1 and θ_2 commensurable. However, our computational experience has revealed that the proposed unscaled strategy seems to work better for the typical practical values of the tolerances ε_1 and ε_2 as delineated in Section 8. Moreover, the chosen values of $\varepsilon_1 < \varepsilon_2$ induce a limited priority-type branching strategy that suitably favors branching on the complementarity restrictions to some extent, which promotes computational effectiveness.

5 A semi-smooth algorithm for QEiCP

We begin by writing the system (11) as follows:

$$\lambda \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} -B & -C \\ I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} w \\ t \end{bmatrix} \tag{22a}$$

$$e^\top y + e^\top x = 1 \tag{22b}$$

$$y_i \geq 0,\ w_i \geq 0,\ y_i w_i = 0,\ i = 1,\ldots,n \tag{22c}$$

$$x_i \geq 0,\ t_i \geq 0,\ x_i t_i = 0,\ i = 1,\ldots,n. \tag{22d}$$

Since $A \in \mathrm{PD}$ and $C \notin S_0$, Theorem 1 implies that $\lambda > 0$ in any solution, whence $\lambda \geq 0$ does not need to be included in the solution. Furthermore, the constraints (22c) and (22d) are a consequence of the complementarity conditions (11c).

It is well known that complementarity constraints can be transformed into equality constraints by using suitable semi-smooth functions. We apply such a transformation to our system of inequalities, using first the *Fischer-Burmeister function* and then the *min function*.

Approach 1 The Fischer-Burmeister function $\varphi_{\mathrm{FB}} : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\varphi_{\mathrm{FB}}(a,b) = a + b - \sqrt{a^2 + b^2}.$$

.

This function satisfies the following relation (see [7]):

$$\varphi_{\mathrm{FB}}(a,b) = 0 \Leftrightarrow a \geq 0,\ b \geq 0,\ ab = 0.$$

As a consequence, the constraints (22c) and (22d), can be replaced by equality constraints, by introducing the functions $\Phi_1 : \mathbb{R}^{2n} \to \mathbb{R}^n$ and $\Phi_2 : \mathbb{R}^{2n} \to \mathbb{R}^n$ defined as

$$\Phi_1(x,t) = \begin{bmatrix} \varphi_{\mathrm{FB}}(x_1,t_1) \\ \vdots \\ \varphi_{\mathrm{FB}}(x_n,t_n) \end{bmatrix} \quad \text{and} \quad \Phi_2(y,w) = \begin{bmatrix} \varphi_{\mathrm{FB}}(y_1,w_1) \\ \vdots \\ \varphi_{\mathrm{FB}}(y_n,w_n) \end{bmatrix}, \tag{23}$$

and by setting

$$\Phi_1(x,t) = 0 \quad \text{and} \quad \Phi_2(y,w) = 0.$$

Thus, the system (22) can be seen as a system of equations

$$\Psi_{\mathrm{FB}}(x,y,w,t,\lambda) = 0 \tag{24}$$

with

$$\Psi_{\mathrm{FB}}(x,y,w,t,\lambda) = \begin{bmatrix} \Phi_1(x,t) \\ \Phi_2(y,w) \\ (\lambda A+B)y+Cx-w \\ \lambda x-y-t \\ e^\top y+e^\top x-1 \end{bmatrix}. \tag{25}$$

Note that (24) is a nonsmooth set of equations, because the function φ is everywhere differentiable except at the origin. However, since Ψ_{FB} is *semi-smooth* (see e.g. [9]), we can apply the *semi-smooth Newton method* for solving (24), as described in the following. Let $(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ be the current iterate, satisfying $e^\top\bar{y}+e^\top\bar{x}=1$. Such a point can be regarded as a solution to the system (22) with tolerances $\varepsilon_1,\varepsilon_2$, if it satisfies

$$\max\{\|\bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x}\|,\|\bar{t}-\bar{\lambda}\bar{x}+\bar{y}\|\}<\varepsilon_1, \max\{\|\Phi_1(\bar{x},\bar{t})\|,\|\Phi_2(\bar{y},\bar{w})\|\}<\varepsilon_2, \tag{26}$$

with Φ_1 and Φ_2 given by (23). If (26) does not hold, then we update the point by applying a Newton iteration, i.e., by computing the *semi-smooth Newton direction* as a solution to the following linear system

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda}) \begin{bmatrix} d_x \\ d_y \\ d_w \\ d_t \\ d_\lambda \end{bmatrix} = \begin{bmatrix} \sqrt{\bar{x}^2+\bar{t}^2}-\bar{x}-\bar{t} \\ \sqrt{\bar{y}^2+\bar{w}^2}-\bar{y}-\bar{w} \\ \bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x} \\ \bar{t}-\bar{\lambda}\bar{x}+\bar{y} \\ 0 \end{bmatrix}, \tag{27}$$

where $GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is the Clarke generalized Jacobian of Ψ_{FB} at $(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$. This matrix can be computed as follows:

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda}) = \begin{bmatrix} \widetilde{E} & 0 & 0 & \widetilde{F} & 0 \\ 0 & \hat{E} & \hat{F} & 0 & 0 \\ C & (\bar{\lambda}A+B) & -I_n & 0 & A\bar{y} \\ \bar{\lambda}I_n & -I_n & 0 & -I_n & \bar{x} \\ e^\top & e^\top & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(4n+1)\times(4n+1)}. \tag{28}$$

Here I_n is the identity matrix of order n, and $\widetilde{E},\widetilde{F},\hat{E},\hat{F}\in\mathbb{R}^{n\times n}$ are diagonal matrices whose diagonal elements are defined as follows:

$$(\widetilde{E}_{ii},\widetilde{F}_{ii}) = \begin{cases} \left(1-\frac{\bar{x}_i}{\sqrt{\bar{x}_i^2+\bar{t}_i^2}}, 1-\frac{\bar{t}_i}{\sqrt{\bar{x}_i^2+\bar{t}_i^2}}\right) & \text{if } (\bar{x}_i,\bar{t}_i)\neq 0 \\ \left(1-\widetilde{\xi}_i, 1-\widetilde{\eta}_i\right) & \text{if } (\bar{x}_i,\bar{t}_i)=0 \end{cases} \quad \forall i=1,\ldots,n, \tag{29}$$

with $\tilde{\xi}_i^2+\tilde{\eta}_i^2=1$,

$$(\hat{E}_{ii},\hat{F}_{ii})=\begin{cases}\left(1-\frac{\bar{y}_i}{\sqrt{\bar{y}_i^2+\bar{w}_i^2}},1-\frac{\bar{w}_i}{\sqrt{\bar{y}_i^2+\bar{w}_i^2}}\right) & \text{if } (\bar{y}_i,\bar{w}_i)\neq 0\\ \left(1-\hat{\xi}_i,1-\hat{\eta}_i\right) & \text{if } (\bar{y}_i,\bar{w}_i)=0\end{cases}\quad \forall i=1,\dots,n, \tag{30}$$

with $\hat{\xi}_i^2+\hat{\eta}_i^2=1$. In practice we use $(\tilde{\xi}_i,\tilde{\eta}_i)=(1,0)$ and $(\hat{\xi}_i,\hat{\eta}_i)=(1,0)$ for all $i=1,\dots,n$.

If $GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is singular, then the algorithm terminates unsuccessfully. Otherwise the search direction $(d_x,d_y,d_w,d_t,d_\lambda)$ is uniquely determined by (27) and a new point is obtained by

$$\tilde{x}=\bar{x}+d_x,\ \tilde{y}=\bar{y}+d_y,\ \tilde{w}=\bar{w}+d_w,\ \tilde{t}=\bar{t}+d_t,\ \text{and } \tilde{\lambda}=\bar{\lambda}+d_\lambda. \tag{31}$$

Note that the new point satisfies $e^\top\tilde{x}+e^\top\tilde{y}=1$, and is thus used in the next iteration of the method.

Approach 2 The complementarity constraints can be also replaced by using the *min function* defined as

$$\varphi_{\min}(a,b)=0, \tag{32}$$

where $\varphi_{\min}:\mathbb{R}^2\to\mathbb{R}$ is the minimum function $\min\{a,b\}$, which is equal to zero if and only if $a\geq 0$, $b\geq 0$, and $ab=0$. The complementarity constraints (22c) and (22d) can be represented by setting to zero the functions $\Phi_1:\mathbb{R}^{2n}\to\mathbb{R}^n$ and $\Phi_2:\mathbb{R}^{2n}\to\mathbb{R}^n$ defined by

$$\Phi_1(x,t)=\begin{bmatrix}\varphi_{\min}(x_1,t_1)\\ \vdots\\ \varphi_{\min}(x_n,t_n)\end{bmatrix}\quad\text{and}\quad \Phi_2(y,w)=\begin{bmatrix}\varphi_{\min}(y_1,w_1)\\ \vdots\\ \varphi_{\min}(y_n,w_n)\end{bmatrix}. \tag{33}$$

Then, the system (22) is equivalent to the following system of equations:

$$\Psi_{\min}(x,y,w,t,\lambda)=0 \tag{34}$$

with

$$\Psi_{\min}(x,y,w,t,\lambda)=\begin{bmatrix}\Phi_1(x,t)\\ \Phi_2(y,w)\\ (\lambda A+B)y+Cx-w\\ \lambda x-y-t\\ e^\top y+e^\top x-1\end{bmatrix}. \tag{35}$$

Since the function $\Psi_{\min}(x,y,w,t,\lambda)$ is semi-smooth, a solution of the system of equations can be found as in the previous approach by applying the semi-smooth Newton method until the following conditions are satisfied:

$$\max\{\|\bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x}\|,\|\bar{t}-\bar{\lambda}\bar{x}+\bar{y}\|\}<\varepsilon_1,\max\{\|\Phi_1(\bar{x},\bar{t})\|,\|\Phi_2(\bar{y},\bar{w})\|\}<\varepsilon_2, \tag{36}$$

where Φ_1 and Φ_2 are given by (33), and $(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is the current iterate satisfying $e^\top \bar{y}+e^\top \bar{x}=1$. At each Newton iteration a new direction is computed via

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})\begin{bmatrix} d_x \\ d_y \\ d_w \\ d_t \\ d_\lambda \end{bmatrix} = \begin{bmatrix} -\min\{x,t\} \\ -\min\{y,w\} \\ \bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x} \\ \bar{t}-\bar{\lambda}\bar{x}+\bar{y} \\ 0 \end{bmatrix}. \tag{37}$$

The Clarke generalized Jacobian $GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is given by

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda}) = \begin{bmatrix} \widetilde{E} & 0 & 0 & \widetilde{F} & 0 \\ 0 & \hat{E} & \hat{F} & 0 & 0 \\ C & (\bar{\lambda}A+B) & -I_n & 0 & A\bar{y} \\ \bar{\lambda}I_n & -I_n & 0 & -I_n & \bar{x} \\ e^\top & e^\top & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(4n+1)\times(4n+1)}, \tag{38}$$

where I_n is the identity matrix of order n, and where $\widetilde{E},\widetilde{F},\hat{E},\hat{F} \in \mathbb{R}^{n\times n}$ are diagonal matrices with the following diagonal elements:

$$(\widetilde{E}_{ii},\widetilde{F}_{ii}) = \begin{cases} (1,0) & \text{if } \bar{x}_i < \bar{t}_i \\ (0,1) & \text{if } \bar{t}_i < \bar{x}_i \\ (\widetilde{v}_i, 1-\widetilde{v}_i) & \text{if } \bar{x}_i = \bar{t}_i \end{cases} \quad \forall i=1,\ldots,n, \tag{39}$$

with $\widetilde{v}_i \in [0,1]$, and where

$$(\hat{E}_{ii},\hat{F}_{ii}) = \begin{cases} (1,0) & \text{if } \bar{y}_i < \bar{w}_i \\ (0,1) & \text{if } \bar{w}_i < \bar{y}_i \\ (\hat{v}_i, 1-\hat{v}_i) & \text{if } \bar{y}_i = \bar{w}_i \end{cases} \quad \forall i=1,\ldots,n, \tag{40}$$

with $\hat{v}_i \in [0,1]$. In practice we use $\widetilde{v}_i = 0$ and $\hat{v}_i = 0$ for all $1,\ldots,n$.

If $GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is singular, then the algorithm terminates unsuccessfully. Otherwise, the direction $(d_x,d_y,d_w,d_t,d_\lambda)$ is uniquely determined by (37) and the new iterate is defined by

$$\tilde{x}=\bar{x}+d_x,\ \tilde{y}=\bar{y}+d_y,\ \tilde{w}=\bar{w}+d_w,\ \tilde{t}=\bar{t}+d_t,\ \text{and } \tilde{\lambda}=\bar{\lambda}+d_\lambda, \tag{41}$$

which satisfies $e^\top\tilde{x}+e^\top\tilde{y}=1$.

Next, we present in a formal way the main steps of the semi-smooth algorithm for solving the system (22), which is valid for both the Fischer-Burmeister and min function approaches.

Algorithm 2 Semi-smooth Newton algorithm

▷ **Step** 0 **(Initialization)**

Let $(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ be an initial point such that $e^\top\bar{y}+e^\top\bar{x}=1$.
Let $\tilde{\varepsilon}_1$ and $\tilde{\varepsilon}_2$ be selected positive tolerances.
Compute Φ_1 and Φ_2 given in (23) (or (33)).
while $(\max\{\|\bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x}\|,\|\bar{t}-\bar{\lambda}\bar{x}+\bar{y}\|\}>\tilde{\varepsilon}_1$ OR $\max\{\|\Phi_1\|,\|\Phi_2\|\}>\tilde{\varepsilon}_2)$ **do**

▷ **Step** 1 **(Newton direction)**

Compute the diagonal matrices $\hat{E},\hat{F},\widetilde{E},\widetilde{F}\in\mathbb{R}^{n\times n}$ given by (30) and (29) (or (40) and (39)).
Compute the Clarke generalized Jacobian GJ at $(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ by using

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})=\begin{bmatrix}\widetilde{E} & 0 & 0 & \widetilde{F} & 0\\ 0 & \hat{E} & \hat{F} & 0 & 0\\ C & (\bar{\lambda}A+B) & -I_n & 0 & A\bar{y}\\ \bar{\lambda}I_n & -I_n & 0 & -I_n & \bar{x}\\ e^\top & e^\top & 0 & 0 & 0\end{bmatrix}$$

if $GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})$ is singular **then**
Stop, and terminate with an unsuccessful termination.
else
Find the semi-smooth Newton direction

$$GJ(\bar{x},\bar{y},\bar{w},\bar{t},\bar{\lambda})\begin{bmatrix}d_x\\ d_y\\ d_w\\ d_t\\ d_\lambda\end{bmatrix}=\begin{bmatrix}-\Phi_1(\bar{x},\bar{t})\\ -\Phi_2(\bar{y},\bar{w})\\ \bar{w}-(\bar{\lambda}A+B)\bar{y}-C\bar{x}\\ \bar{t}-\bar{\lambda}\bar{x}+\bar{y}\\ 0\end{bmatrix}$$

with Φ_1 and Φ_2 given in (23) (or (33)).

▷ **Step** 3 **(Update)**

Compute the new point

$$\tilde{x}=\bar{x}+d_x,\ \tilde{y}=\bar{y}+d_y,\ \tilde{w}=\bar{w}+d_w,\ \tilde{t}=\bar{t}+d_t,\ \text{ and }\tilde{\lambda}=\bar{\lambda}+d_\lambda$$

and let $\bar{x}=\tilde{x},\bar{y}=\tilde{y},\ \bar{w}=\tilde{w},\ \hat{t}=\tilde{t},$ and $\bar{\lambda}=\tilde{\lambda}$.
end if
end while

If the algorithm terminates with success, then $\bar{\lambda}$ is a quadratic complementarity eigenvalue, and $(1+\bar{\lambda})\bar{x}$ is a corresponding quadratic complementarity eigenvector.

6 A hybrid algorithm for QEiCP

As discussed in [10], the enumerative algorithm is globally convergent to a solution of QEiCP. However, in many cases, the algorithm is only able to terminate with a near-solution to QEiCP. On the other hand, the semi-smooth method is a fast local algorithm, but lacks the global convergence feature. Hence, we can combine the good features of both the algorithms in a hybrid method based on the same ideas of a similar procedure discussed in [10]. The steps of the hybrid method are presented below.

Algorithm 3 Hybrid algorithm

▷ **Step** 0 **(Initialization)**

Let $\bar{\varepsilon}_1$ and $\bar{\varepsilon}_2$ be two positive tolerances for switching from the enumerative method to the semi-smooth and, let ε_1 and ε_2 be the tolerances used in Algorithm 1, such that $\varepsilon_1 < \bar{\varepsilon}_1$ and $\varepsilon_2 < \bar{\varepsilon}_2$.

Let *nmaxit* be the maximum number of iterations allowed to be performed by the semi-smooth method.

▷ **Step** 1 **(Decision step)**

Let $(\bar{x}, \bar{y}, \bar{v}, \bar{w}, \bar{\lambda})$ be the stationary point associated with the node k and compute θ_1 and θ_2 in (20) and (21), respectively.

while $(\theta_1 > \varepsilon_1$ OR $\theta_2 > \varepsilon_2)$ **do**
    if $(\theta_1 \leq \bar{\varepsilon}_1$ AND $\theta_2 \leq \bar{\varepsilon}_2)$ **then**
        Apply Algorithm 2.
        if Algorithm 2 terminates with a solution $(x^*, y^*, w^*, t^*, \lambda^*)$ **then**
            Stop; set $\bar{\lambda} = \lambda^*$ and $\bar{x} = x^*$.
        else if GJ$(\bar{x}, \bar{y}, \bar{w}, \bar{t}, \bar{\lambda})$ is singular OR **if** the number of iterations is equal to *nmaxit* **then**
            Apply Steps 2 and 3 of Algorithm 1 continuing with the node k and the solution $(\bar{x}, \bar{y}, \bar{v}, \bar{w}, \bar{\lambda})$ given at the beginning of this step.
            Compute θ_1 and θ_2 in (20) and (21), respectively.
        end if
    else
        Apply Steps 2 and 3 of Algorithm 1.
        Compute θ_1 and θ_2 in (20) and (21), respectively.
    end if
end while

7 Computing a positive complementary eigenvalue for EiCP

Consider the EiCP (1)–(4). In this section, we address the problem of the existence and computation of a positive complementary eigenvalue λ for this EiCP. In practice, such a demand occurs quite often [19]. If EiCP is symmetric, i.e., B and C are symmetric matrices (B is PD), then the problem can be solved as in [21]. Hence, we consider the asymmetric case, where at least one of the matrices B or C is not symmetric. Furthermore, we consider the following classes of matrices:

(i) C is a V-matrix ($C \in V$) if and only if there exists an $\bar{x} > 0$ such that $\bar{x}^\top C\bar{x} > 0$
(ii) C is an S-matrix ($C \in S$) if and only if there exists an $\bar{x} \geq 0$ such that $C\bar{x} > 0$.

The following properties, proved in [6], hold between the above classes and the PD and S_0 classes:

Theorem 6 *(i)* $C \in PD \Rightarrow C \in S \Rightarrow C \in V$.
(ii) $C \in S \Leftrightarrow -C^\top \notin S_0$.

Due to this property, verifying that C is not an S-matrix reduces to solving a linear program. Furthermore, showing that a matrix $C \in V$ is equivalent to showing that the following nonlinear program has a feasible point $\bar{x}$ with $f(\bar{x}) > 0$:

$$\begin{array}{ll} \text{Maximize} & \frac{1}{2}x^\top(C+C^\top)x = f(x) \\ \text{subject to} & e^\top x = 1 \\ & x \geq 0, \end{array}$$

where e is a conformable vector of ones. Despite this problem being NP-hard [15], it is in many cases very easy to verify whether a matrix belongs to the class V [11], [21].

Moreover, the following result has been established in [21]:

Theorem 7 *If C is symmetric and $B \in PD$, then EiCP(B,C) has a positive eigenvalue if and only if $C \in V$.*

Furthermore, such a positive complementary eigenvalue can be computed by applying an ascent nonlinear programming algorithm with an initial point $\bar{x}$ satisfying $\bar{x}^\top C\bar{x} > 0$ in order to find a stationary point to the quadratic fractional programming (5) (see [11], [21]).

Consider now the case of B or C or both being symmetric. Then $C \in V$ is still a necessary condition for a positive complementary eigenvalue for EiCP, but it is no longer sufficient. In fact,

$$C = \begin{bmatrix} 2 & -3 \\ 1 & -1 \end{bmatrix} \in V$$

and the EiCP(B,C) with B being the identity matrix has no positive complementary eigenvalue. Theorems 1 and 6 provide a sufficient condition for the existence of such an eigenvalue, as the following result holds:

Theorem 8 *If $B \in PD$ and $C^\top \in S$, then EiCP(B,C) has a positive complementary eigenvalue.*

Proof If $C^\top \in S$, then by Theorem 6, $-C \notin S_0$. Since $B \in$ PD, then QEiCP($B,0,-C$) has a positive (and a negative) eigenvalue. Hence, $\lambda = \mu^2$ is a positive complementary eigenvalue of EiCP. □

This condition is sufficient, but not necessary. In fact for the following matrices:

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; C = \begin{bmatrix} 1 & -2 \\ -3 & 0 \end{bmatrix},$$

the EiCP(B,C) has a positive complementary eigenvalue, but $C^\top \notin S$.

The following example shows that an EiCP(B,C) with $B \in$ PD and $C^\top \in S$ may have a negative eigenvalue. Consider the matrices

$$B = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \; C = \begin{bmatrix} -1 & 1 \\ \frac{1}{2} & 1 \end{bmatrix}.$$

Then it is easy to see that $B \in$ PD and $C^\top \in S$. By Theorem 8, EiCP(B,C) has at least a positive complementary eigenvalue. However, this EiCP also has the negative complementary eigenvalue $\lambda = -1$. If we apply an ordinary algorithm to compute a solution to the EiCP, then this procedure may find the negative eigenvalue. Instead of solving the EiCP(B,C) directly, it is more advisable to find a solution to QEiCP($B, 0, -C$) in order to guarantee the computation of a positive complementary eigenvalue λ for EiCP(B,C), that is, to find $\lambda = \mu^2 > 0$, with μ being the quadratic complementary eigenvalue computed by the hybrid enumerative algorithm discussed in this paper.

8 Computational experience

In this section, we discuss the numerical performance of the proposed algorithms for computing quadratic complementary eigenvalues. The enumerative algorithm has been implemented in MATLAB [16] and the IPOPT (Interior Point OPTimizer) solver [27] has been used to find a (local) solution to the nonlinear problem $\mathbf{NLP}_4$(k) in (18) at each node k.

We consider two sets of test problems with $A \in$ PD and $C \notin S_0$. The first set of problems, called Test Problems 1, deal with co-regular and co-hyperbolic QEiCPs, which always have a solution. The matrices A and $-C$ were both chosen as the identity matrix, while the matrix B was randomly generated with elements uniformly distributed in the intervals $[0,1]$, $[0,10]$, $[0,100]$, and $[0,300]$. These problems are denoted by $\text{RAND}(0,m,n)$, where 0 and m are the end-points of the interval, and n represents the dimension of the problem, i.e., the matrices $A, B, C \in \mathbb{R}^{n\times n}$. We have considered $n = 3, 5, 10, 20, 30, 50,$ and 100. For the second test, called Test Problems 2, $C \notin S_0$ was chosen such that the resulting QEiCP is not co-hyperbolic. In particular, C has the following structure:

$$C = \begin{bmatrix} -E & -h \\ -g^\top & c_{nn} \end{bmatrix},$$

where $E \in \mathbb{R}^{(n-1)\times(n-1)}$ is a square matrix with randomly generated elements in the interval $[0,m]$, $h \in \mathbb{R}^{n-1}$ and $g \in \mathbb{R}^{n-1}$ are vectors with randomly generated elements in the same interval, and the element $c_{nn} = (m/2)^2 + 1$. The matrices A and B were chosen as in the first case.

Problem	λ	f	l	u	Nodes	CPU
RAND(0, 1, 3)	0.540591	8.55776e-17	0.288722	7.242641	0	9.43264e-02
RAND(0, 1, 5)	0.579740	1.63562e-16	0.250875	12.071068	0	1.08970e-01
RAND(0, 1, 10)	0.353326	2.54107e-16	0.121939	24.142136	0	1.17104e-01
RAND(0, 1, 20)	0.216038	2.72803e-15	0.076108	48.284271	0	1.77654e-01
RAND(0, 1, 30)	0.152153	9.71651e-12	0.049863	72.426407	0	2.99098e-01
RAND(0, 1, 50)	0.071278	2.31483e-10	0.030815	120.710678	0	1.02978e+00
RAND(0, 1, 100)	0.029856	2.32097e-09	0.017024	241.421356	0	2.94410e+00
RAND(0, 10, 3)	0.064433	6.17703e-16	0.044192	7.242641	0	3.71968e-02
RAND(0, 10, 5)	0.181341	7.09644e-15	0.032618	12.071068	0	8.36979e-02
RAND(0, 10, 10)	0.031470	5.08357e-08	0.014044	24.142136	0	2.00814e-01
RAND(0, 10, 20)	0.037200	4.74654e-14	0.008171	48.284271	0	7.75963e-01
RAND(0, 10, 30)	0.021879	9.05131e-08	0.005807	72.426407	0	4.16905e-01
RAND(0, 10, 50)	0.005521	1.44288e-09	0.003487	120.710678	0	9.68685e-01
RAND(0, 10, 100)	0.004779	1.41496e-09	0.001826	241.421356	0	2.89875e+00
RAND(0, 100, 3)	0.006558	4.76544e-12	0.005080	7.242641	0	1.06655e-01
RAND(0, 100, 5)	0.004492	1.13846e-11	0.002625	12.071068	0	2.52097e-01
RAND(0, 100, 10)	0.002790	5.87198e-11	0.001509	24.142136	0	1.78784e-01
RAND(0, 100, 20)	0.010015	2.01563e-09	0.000745	48.284271	0	1.81592e-01
RAND(0, 100, 30)	0.005434	1.29756e-08	0.000557	72.426407	0	6.95276e-01
RAND(0, 100, 50)	0.005470	1.07865e-08	0.000335	120.710678	0	4.45850e+00
RAND(0, 100, 100)	0.001888	2.29598e-09	0.000177	241.421356	0	9.19532e+00
RAND(0, 300, 3)	0.002392	2.27355e-11	0.002198	7.242641	0	8.54462e-02
RAND(0, 300, 5)	0.001428	5.69616e-11	0.001060	12.071068	0	1.19519e-01
RAND(0, 300, 10)	0.000664	1.20981e-10	0.000550	24.142136	0	1.68899e-01
RAND(0, 300, 20)	0.000647	8.75978e-09	0.000273	48.284271	0	1.50398e-01
RAND(0, 300, 30)	0.000671	2.67442e-08	0.000164	72.426407	0	6.04931e-01
RAND(0, 300, 50)	0.000622	9.60127e-09	0.000115	120.710678	1	8.93506e-01
RAND(0, 300, 100)	0.001598	6.58115e-08	0.000060	241.421356	1	4.03082e+01

Table 1 Performance of the enumerative method for solving Test Problems 1.

8.1 Performance of the enumerative method

Tables 1 and 2 report the computational experience when solving Test Problems 1 and 2, respectively. The enumerative method was run with the tolerances $\varepsilon_1 = 10^{-5}$ and $\varepsilon_2 = 10^{-4}$. In these tables, we have reported the computed value of the eigenvalue, the value of the function f derived at the solution, the value of the lower and upper bounds computed as in Section 3, the number of nodes enumerated by the algorithm, and the CPU time in seconds. The symbol * indicates that the enumerative algorithm was not able to solve the problem, i.e., the algorithm attained the maximum number of iterations, fixed as $n_{\max} = 500$. In this case we include the value of the objective function for the best stationary point. The value zero in the column titled "Nodes" indicates that a solution to QEiCP was found at the root node itself. Note that the greater computational effort, i.e., the larger number of explored nodes, in solving Test Problems 2 is due to the more complex structure of the matrix C.

As a benchmark for comparison, we solved these same problems using BARON (Branch-And-Reduce Optimization Navigator; see [26]), which is an optimization solver for the global solution of algebraic nonlinear programs and mixed-integer nonlinear programs. This software package implements a branch-and-cut algorithm, enhanced with a variety of constraint propagation and duality techniques for reducing ranges of variables in the course of the algorithm. The code for solving the nonlinear problem $\mathbf{NLP}_1$ given in (12) for both Test Problems 1 and 2 was implemented in the General Algebraic Modeling Systems (GAMS) language (see [5]) and the solver

Problem	λ	f	l	u	Nodes	CPU
RAND(0, 1, 3)	0.553862	6.14338e-15	0.290421	8.684863	6	3.08398e-01
RAND(0, 1, 5)	0.820148	1.36299e-16	0.435208	20.763329	0	1.22634e-01
RAND(0, 1, 10)	0.703165	5.17996e-15	0.391192	60.947348	11	3.04822e+00
RAND(0, 1, 20)	1.157398	1.81742e-14	0.468243	262.834609	89	1.24254e+01
RAND(0, 1, 30)	0.987340	1.83278e-10	0.463858	576.074031	0	1.66142e+00
RAND(0, 1, 50)	1.077929	6.24270e-09	0.487771	1600.322504	5	1.48014e+01
RAND(0, 1, 100)	1.067021	3.50061e-07	0.481082	6107.568272	34	2.20742e+02
RAND(0, 10, 3)	1.440810	1.11071e-16	0.461292	70.978749	0	4.24581e-02
RAND(0, 10, 5)	1.715031	1.35236e-15	0.511997	186.674901	1	1.42548e-01
RAND(0, 10, 10)	0.899639	5.72321e-13	0.406875	577.396692	19	3.12358e+00
RAND(0, 10, 20)	1.922596	1.22525e-13	0.472350	2427.677751	28	1.98347e+01
RAND(0, 100, 30)	4.313552	4.09191e-11	0.472373	53597.732297	66	2.11926e+02
RAND(0, 10, 50)	1.432682	2.80619e-11	0.477837	15390.665702	32	1.59750e+02
RAND(0, 10, 100)	1.786507	7.19200e-12	0.477161	60715.982034	34	5.00920e+02
RAND(0, 100, 3)	0.537246	1.91113e-16	0.289377	347.188118	0	6.04616e-02
RAND(0, 100, 5)	1.078916	1.02845e-12	0.387015	1377.115452	20	1.48914e+00
RAND(0, 100, 10)	1.161560	1.02364e-11	0.443875	6175.750686	23	1.47605e+01
RAND(0, 100, 20)	1.760194	2.31492e-09	0.474284	24220.474672	33	5.53303e+01
RAND(0, 100, 30)	1.231730	5.09885e-09	0.472373	53597.732297	261	2.74036e+02
RAND(0, 100, 50)	1.359856	6.97896e-09	0.481880	152454.077061	77	4.26262e+02
RAND(0, 100, 100)	1.081376	4.76065e-05	0.479115	601687.297576	173	6.17715e+03
RAND(0, 300, 3)	1.100964	9.26871e-17	0.452189	1736.939519	17	9.45302e-01
RAND(0, 300, 5)	0.814907	1.63479e-08	0.403250	3645.329574	24	5.98033e+00
RAND(0, 300, 10)	4.927159	7.82831e-13	0.441231	18197.076729	24	4.83416e+01
RAND(0, 300, 20)	2.295587	2.25558e-06	0.451253	71033.543172	72	1.28525e+02
RAND(0, 300, 30)	1.310145	1.80424e-08	0.467756	164262.282508	76	1.44973e+02
RAND(0, 300, 50)	*			[4.05705e-01]		
RAND(0, 300, 100)	*			[2.48154e-01]		

Table 2 Performance of the enumerative method for solving Test Problems 2.

BARON was used with default options. The numerical results for Test Problems 1 are shown in Table 3, while those for Test Problems 2 are displayed in Table 4. We use the notation * to indicate that BARON was not able to find a solution to QEiCP.

Comparing Tables 2 and 4, we see that the enumerative algorithm fails only two times in finding a solution versus seven times for BARON. Moreover, the computational time for the enumerative method was comparable and in general smaller than that required by BARON.

8.2 Performance of the semi-smooth method

The same test problems were solved by using the semi-smooth Newton algorithm; the complementarity constraints were represented by using both the Fischer-Burmeister function and the min function (see Section 5). Tables 5 and 6 present the results for Test Problems 1 and 2, respectively. The starting point was chosen as $\bar{\lambda} = 1$, $(\bar{x}, \bar{y}) = (1/2n, \ldots, 1/2n)$, $\bar{w} = (A\bar{\lambda} + B)\bar{y} + C\bar{x}$, $\bar{t} = \bar{\lambda}\bar{x} - \bar{y}$. It is well-known that the semi-smooth Newton algorithm is very sensitive to the choice of the starting point. Thus, numerical experiments were also performed where the vertices of the simplex were taken as starting points. In this particular case, the performance of the algorithm turned out to be similar for all choices of the starting point, and hence we have only reported the results for the first choice. In Tables 5 and 6, we report the value of the computed eigenvalue, the number of iterations taken by the algorithm to converge,

Problem	λ	f	CPU
RAND(0, 1, 3)	0.580550	3.49656e-15	3.86000e-01
RAND(0, 1, 5)	0.604048	8.56266e-12	1.50000e-01
RAND(0, 1, 10)	0.638287	8.21710e-14	2.00000e-01
RAND(0, 1, 20)	0.670098	1.76270e-13	1.62000e-01
RAND(0, 1, 30)	0.688537	7.13992e-14	1.91000e-01
RAND(0, 1, 50)	0.492137	1.59410e-11	2.00000e-01
RAND(0, 1, 100)	0.020337	5.83765e-11	3.39100e+00
RAND(0, 10, 3)	0.116253	1.93421e-12	1.84000e-01
RAND(0, 10, 5)	0.071066	1.78546e-13	1.67000e-01
RAND(0, 10, 10)	0.113500	6.48410e-14	2.29000e-01
RAND(0, 10, 20)	0.105702	2.40212e-10	1.63000e-01
RAND(0, 10, 30)	0.099982	4.04072e-12	1.93000e-01
RAND(0, 10, 50)	0.087719	1.86553e-13	1.37000e-01
RAND(0, 10, 100)	0.004859	1.58826e-10	1.53500e+00
RAND(0, 100, 3)	0.020546	1.23185e-16	1.75000e-01
RAND(0, 100, 5)	0.013835	3.51415e-19	1.73000e-01
RAND(0, 100, 10)	0.021313	1.04600e-14	1.53000e-01
RAND(0, 100, 20)	0.012589	1.56146e-15	1.61000e-01
RAND(0, 100, 30)	0.008782	3.41173e-10	2.38000e-01
RAND(0, 100, 50)	0.007166	4.50590e-10	3.17000e-01
RAND(0, 100, 100)	0.000357	5.02497e-10	7.02000e-01
RAND(0, 300, 3)	0.003900	4.55744e-18	1.47000e-01
RAND(0, 300, 5)	0.004200	1.37019e-23	1.42000e-01
RAND(0, 300, 10)	0.004485	1.50219e-12	1.49000e-01
RAND(0, 300, 20)	0.002689	7.00246e-12	2.41000e-01
RAND(0, 300, 30)	0.002670	5.06612e-10	1.76000e-01
RAND(0, 300, 50)	0.005649	6.26275e-13	1.55000e-01
RAND(0, 300, 100)	0.000205	3.09113e-10	6.45100e+00

Table 3 Performance of BARON for solving Test Problems 1.

and the CPU time in seconds. The notation “GJ singular” indicates that the algorithm terminated unsuccessfully with the singularity of the Clarke generalized Jacobian.

Tables 5 and 6 also provide a comparison for the performance of the algorithm when using the Fischer-Burmeister function versus the min function for representing the complementarity constraints. If we consider the number of times that a solution was found, the use of the min function seems to be preferable for Test Problems 1, while the use of the FB function works better in solving Test Problems 2.

Note that the semi-smooth method is faster than the enumerative algorithm for obtaining a solution, but on the other hand, it often terminates unsuccessfully with the singularity of the Generalized Jacobian.

8.3 Performance of the hybrid method

For all the test problems for which the enumerative method required more than one node for finding a solution, we applied the hybrid method proposed in Section 6. This algorithm was implemented by using both the Fisher-Burmeister and the min functions. The values of the tolerances $\overline{\varepsilon_1}$ and $\overline{\varepsilon_2}$ used to switch from the enumerative method to the semi-smooth Newton method were both set to 10^{-1}. For the semi-smooth Newton algorithm, the values of the tolerances to terminate the algorithm were taken as $\varepsilon_1 = 10^{-6}$ and $\varepsilon_2 = 10^{-6}$. The maximum number of iterations for the semi-smooth method was fixed as 100. The results for Test Problems 1 and 2 are sum-

Problem	λ	f	CPU
RAND(0, 1, 3)	0.553862	1.16923e-13	4.43000e-01
RAND(0, 1, 5)	0.820147	1.84665e-11	1.53000e-01
RAND(0, 1, 10)	0.703152	5.83075e-11	2.00000e-01
RAND(0, 1, 20)	1.157398	7.97059e-17	7.26500e+00
RAND(0, 1, 30)	0.987340	1.38197e-19	4.72100e+00
RAND(0, 1, 50)	*		
RAND(0, 1, 100)	*		
RAND(0, 10, 3)	1.440809	5.73612e-13	1.07100e+00
RAND(0, 10, 5)	1.713643	1.04554e-13	7.15000e-01
RAND(0, 10, 10)	0.879084	1.54890e-11	1.96000e-01
RAND(0, 10, 20)	1.922596	1.79492e-18	1.22591e+02
RAND(0, 10, 30)	0.967767	6.21467e-20	9.28900e+00
RAND(0, 10, 50)	1.591280	3.48178e-10	3.88145e+02
RAND(0, 10, 100)	*		
RAND(0, 100, 3)	0.805417	1.15432e-13	3.64000e-01
RAND(0, 100, 5)	1.597662	4.23887e-12	1.71000e-01
RAND(0, 100, 10)	2.386302	4.04346e-12	3.62000e-01
RAND(0, 100, 20)	1.751899	1.84497e-10	4.96000e-01
RAND(0, 100, 30)	1.218751	4.92004e-23	6.54917e+02
RAND(0, 100, 50)	*		
RAND(0, 100, 100)	*		
RAND(0, 300, 3)	1.100964	1.33835e-27	2.43300e+00
RAND(0, 300, 5)	0.814898	3.07773e-14	1.87000e-01
RAND(0, 300, 10)	1.325918	4.43267e-11	9.51000e-01
RAND(0, 300, 20)	1.684025	3.50538e-10	5.38800e+00
RAND(0, 300, 30)	2.043501	7.55092e-10	7.33890e+02
RAND(0, 300, 50)	*		
RAND(0, 300, 100)	*		

Table 4 Performance of BARON for solving Test Problems 2.

marized in Tables 7 and 8, respectively, where we report the value of the computed eigenvalue, the number of times that the semi-smooth Newton method was called, which we indicate as “Ntime”, the number of nodes enumerated by the algorithm, and the CPU time in seconds. The symbol * indicates that the use of the semi-smooth Newton method was not helpful in finding a solution.

We observe that the additional use of the semi-smooth Newton method allows us to find a solution by enumerating a fewer number of nodes. For nine problems, the semi-smooth method with the use of the Fischer-Burmeister function was called only once. This happens seven times when the min function is chosen. However, even when the hybrid method solves both the minimization problem $\mathbf{NLP}_4$(k) and applies the semi-smooth method for some k, in general, the performance in terms of CPU time improves.

We also note that the use of the hybrid method was not helpful in finding a solution for five problems by using the Fischer-Burmeister function and in four cases with the use of the min function. Moreover the min function was not able to solve two problems within the given number of iterations, while this situation does not occur for the Fischer-Burmeister function. So in general, the Fischer-Burmeister function appears to perform better than the min function.

We conclude that the hybrid method with the Fischer-Burmeister function improves over the enumerative method and is recommended in practice for solving the QEiCP with $A \in$ PD and $C \notin S_0$ via the equivalent EiCP.

	FB function			min function		
Problem	λ	niter	CPU	λ	niter	CPU
RAND(0, 1, 3)	0.540591	11	1.79230e-02	0.747744	8	1.16965e-02
RAND(0, 1, 5)	0.832908	42	1.01189e-02	0.409159	7	1.30902e-03
RAND(0, 1, 10)	0.305448	12	4.54200e-03	0.231762	7	1.77044e-03
RAND(0, 1, 20)	*	GJ singular		0.108348	7	3.16456e-03
RAND(0, 1, 30)	0.483898	55	6.20506e-02	0.075831	6	5.60512e-03
RAND(0, 1, 50)	*	GJ singular		*	GJ singular	
RAND(0, 1, 100)	0.043377	37	3.33108e-01	*	GJ singular	
RAND(0, 10, 3)	0.084907	17	5.12231e-03	0.054726	5	8.87787e-04
RAND(0, 10, 5)	0.045114	20	4.22854e-03	0.942878	7	1.02378e-03
RAND(0, 10, 10)	0.028512	26	7.31569e-03	0.082063	6	2.07664e-03
RAND(0, 10, 20)	*	GJ singular		0.037174	13	6.32527e-03
RAND(0, 10, 30)	*	GJ singular		0.020982	6	4.06817e-03
RAND(0, 10, 50)	*	GJ singular		0.009154	5	8.86932e-03
RAND(0, 10, 100)	*	GJ singular		*	GJ singular	
RAND(0, 100, 3)	*	GJ singular		0.008239	6	8.56142e-04
RAND(0, 100, 5)	0.017293	31	5.93911e-03	0.023631	5	8.67260e-04
RAND(0, 100, 10)	*	GJ singular		0.001904	6	1.49119e-03
RAND(0, 100, 20)	*	GJ singular		0.000998	6	2.73863e-03
RAND(0, 100, 30)	0.000663	16	1.44868e-02	0.001345	5	3.60375e-03
RAND(0, 100, 50)	0.000406	16	3.86470e-02	*	GJ singular	
RAND(0, 100, 100)	*	GJ singular		*	GJ singular	
RAND(0, 300, 3)	0.002543	35	7.36188e-03	0.002386	6	8.47589e-04
RAND(0, 300, 5)	*	GJ singular		0.002880	5	1.12000e-03
RAND(0, 300, 10)	*	GJ singular		0.000639	6	1.59596e-03
RAND(0, 300, 20)	0.000339	17	9.84050e-03	0.000339	5	2.25753e-03
RAND(0, 300, 30)	0.000217	16	1.47687e-02	0.000374	5	4.27258e-03
RAND(0, 300, 50)	*	GJ singular		0.000289	5	8.31467e-03
RAND(0, 300, 100)	*	GJ singular		*	GJ singular	

Table 5 Performance of the semi-smooth Newton method for solving Test Problems 1.

As discussed before, the algorithm always find a positive quadratic complementary eigenvalue for QEiCP. If we are interested in a negative eigenvalue, then the matrix H should be used instead of the matrix G in the $2n$-dimensional EiCP. The algorithmic process is similar with B replaced by $-B$.

8.4 Computing a positive eigenvalue for EiCP

We present the numerical performance of the arguments presented in Section 7 for computing a positive complementary eigenvalue λ for the EiCP (1)–(4). The enumerative and the hybrid methods proposed in this paper are applied for solving the QEiCP$(B,0,-C)$ where B is the identity matrix and

$$C = \begin{bmatrix} 1 & e^\top \\ g & H \end{bmatrix}$$

where $e \in \mathbb{R}^{n-1}$ is a vector of ones, $H = \text{RAND}(0,m,n-1) - (m+1)\, I_{n-1}$ with I_{n-1} being the identity matrix of order $n-1$, and $g \in \mathbb{R}^{n-1}$ is a null vector. Note that $B \in$ PD and $-C \notin S_0$, then QEiCP$(B,0,-C)$ has a solution with $\lambda > 0$ and by Theorem 8 the EiCP(B,C) has a positive complementary eigenvalue equal to λ^2.

Tables 9 and 10 report the computational experience when solving QEiCP$(B,0,-C)$ by the enumerative and the hybrid methods with the same values of tolerances used

	FB function			min function		
Problem	λ	niter	CPU	λ	niter	CPU
RAND(0, 1, 3)	0.553862	10	2.72663e-02	0.553862	20	2.97779e-02
RAND(0, 1, 5)	0.820148	7	1.83033e-03	0.820148	4	1.35607e-03
RAND(0, 1, 10)	0.703165	20	8.45372e-03	*	GJ singular	
RAND(0, 1, 20)	*	GJ singular		1.157398	18	1.71405e-02
RAND(0, 1, 30)	1.033856	26	5.98098e-02	1.033856	42	4.69800e-02
RAND(0, 1, 50)	1.158910	97	4.71429e-01	*	GJ singular	
RAND(0, 1, 100)	*	GJ singular		*	GJ singular	
AND(0, 10, 3)	1.440810	7	2.79211e-03	1.440810	7	1.17902e-03
RAND(0, 10, 5)	1.713642	7	1.47239e-03	1.713642	5	7.19302e-04
RAND(0, 10, 10)	0.864754	5	3.03672e-03	0.864754	4	1.61223e-03
RAND(0, 10, 20)	*	GJ singular		*	GJ singular	
RAND(0, 10, 30)	*	GJ singular		*	GJ singular	
RAND(0, 10, 50)	*	GJ singular		*	GJ singular	
RAND(0, 10, 100)	*	GJ singular		*	GJ singular	
AND(0, 100, 3)	0.537246	11	2.76645e-03	*	GJ singular	
RAND(0, 100, 5)	0.715387	4	1.02165e-03	0.715387	4	6.18377e-04
RAND(0, 100, 10)	1.614171	7	2.23232e-03	1.614171	6	1.53910e-03
RAND(0, 100, 20)	*	GJ singular		*	GJ singular	
RAND(0, 100, 30)	*	GJ singular		*	GJ singular	
RAND(0, 100, 50)	*	GJ singular		*	GJ singular	
RAND(0, 100, 100)	*	GJ singular		*	GJ singular	
AND(0, 300, 3)	1.100964	14	3.57641e-03	1.100963	6	2.93524e-02
RAND(0, 300, 5)	0.814898	5	1.10761e-03	0.814898	4	4.84952e-04
RAND(0, 300, 10)	1.513050	7	3.52424e-03	1.513050	6	1.60881e-03
RAND(0, 300, 20)	*	GJ singular		*	GJ singular	
RAND(0, 300, 30)	*	GJ singular		*	GJ singular	
RAND(0, 300, 50)	*	GJ singular		*	GJ singular	
RAND(0, 300, 100)	*	GJ singular		*	GJ singular	

Table 6 Performance of the semi-smooth Newton method for solving Test Problems 2.

	FB function				min function			
Problem	λ	Ntime	Nodes	CPU	λ	Ntime	Nodes	CPU
RAND(0, 300, 50)	0.005113	1	0	7.71793e-01	0.005113	1	0	6.99095e-01
RAND(0, 300, 100)	0.013423	1	0	3.99891e+01	0.013423	1	0	3.95370e+01

Table 7 Performance of hybrid method for solving Test Problems 1.

for the above test problems. Also in this case, the use of the hybrid method largely reduces the number of iterations necessary to find a solution and it is greatly recommended for computing positive eigenvalues of EiCP.

9 Conclusions

In this paper, we have proposed a hybrid method for solving the Quadratic Eigenvalue Complementarity Problem QEiCP(A,B,C) (6)–(9) when A is a PD matrix and C is not an S_0-matrix. These hypotheses seem to be quite realistic in practice. The algorithm combines a tree search enumerative method with a fast and local semi-smooth Newton algorithm. The method can also be applied to compute a positive eigenvalue of the EiCP(B,C) (1)–(4) when $B \in$ PD and $C^\top \in S$, i.e., $-C \notin S_0$. Computational experience shows that the hybrid enumerative algorithm is quite efficient for solving the QEiCP. As discussed in [4], the use of such an approach for QEiCP with other cones, different from $\mathbb{R}^n_+$, is certainly an interesting subject of future research. Furthermore,

	FB function				min function			
Problem	λ	Ntime	Nodes	CPU	λ	Ntime	Nodes	CPU
RAND(0, 1, 3)	0.553862	1	0	2.13562e-01	0.553862	1	0	3.74785e-01
RAND(0, 1, 10)	0.703165	1	0	5.46687e-01	0.703165	9	8	2.38656e+00
RAND(0, 1, 20)	1.157398	1	0	4.66968e-01	1.157398	1	0	4.27643e-01
RAND(0, 1, 50)	1.088287	1	0	4.90474e+00	1.158910	5	4	1.30513e+01
RAND(0, 1, 100)	1.067021	34	*34	2.64755e+02	1.067021	34	*34	2.62171e+02
RAND(0, 10, 5)	1.715031	1	0	8.62953e-02	1.715031	1	0	7.20144e-02
RAND(0, 10, 10)	0.899637	1	0	1.89607e-01	0.864754	1	0	1.91404e-01
RAND(0, 10, 20)	2.600551	22	22	1.86157e+01	2.600551	8	7	7.29273e+00
RAND(0, 10, 30)	0.967767	20	19	3.57694e+01	1.055296	17	16	2.93933e+01
RAND(0, 10, 50)	1.432682	31	*32	1.78146e+02	1.432682	31	*32	1.78071e+02
RAND(0, 10, 100)	1.786507	33	*34	5.31730e+02	1.786507	33	*34	5.42240e+02
RAND(0, 100, 5)	1.078917	1	0	1.75172e-01	1.597674	1	0	1.66163e-01
RAND(0, 100, 10)	1.762110	1	0	8.72616e-01	1.573841	1	0	8.68189e-01
RAND(0, 100, 20)	1.760173	28	32	5.96798e+01	1.496242	2	1	8.66096e+00
RAND(0, 100, 30)	6.665640	24	26	7.90853e+01	6.665640	24	26	8.74352e+01
RAND(0, 100, 50)	3.077892	31	36	2.41082e+02	3.077891	31	36	2.79176e+02
RAND(0, 100, 100)	1.081376	156	*173	6.32093e+03	1.081376	156	*173	6.39869e+03
RAND(0, 300, 3)	1.100964	6	9	6.00009e-01	1.100964	6	9	5.31684e-01
RAND(0, 300, 5)	0.814907	12	*24	6.07113e+00	0.814898	1	0	3.85505e-01
RAND(0, 300, 10)	1.369286	1	0	2.82615e+00	1.513050	10	9	2.32130e+01
RAND(0, 300, 20)	1.247195	36	67	1.22568e+02	1.319769	1	1	3.57244e+00
RAND(0, 300, 30)	1.309542	35	48	1.32772e+02	1.309542	35	48	1.32300e+02
RAND(0, 300, 50)	1.311051	150	267	1.16466e+03	*			[4.05705e-01]
RAND(0, 300, 100)	1.303152	327	395	9.03701e+03	*			[2.48154e-01]

Table 8 Performance of hybrid method for solving Test Problems 2.

Problem	λ	f	l	u	Nodes	CPU
RAND(0, 1, 3)	1.000008	1.24565e-10	1.000000	8.884124	0	2.00879e-01
RAND(0, 1, 5)	1.000029	6.11137e-10	1.000000	19.733040	0	7.12639e-02
RAND(0, 1, 10)	1.521117	9.71181e-15	1.000000	67.985335	0	1.17073e-01
RAND(0, 1, 20)	2.656284	1.10049e-15	1.000000	251.204372	0	4.49557e-01
RAND(0, 1, 30)	3.493475	1.03825e-13	1.000000	560.502976	0	4.53038e-01
RAND(0, 1, 50)	4.752954	7.62902e-10	1.000000	1561.188930	0	3.40887e+00
RAND(0, 1, 100)	6.907058	3.76504e-08	1.000000	6155.036696	1	3.12231e+01
RAND(0, 10, 3)	1.000009	1.24475e-10	1.000000	22.025204	0	7.04963e-02
RAND(0, 10, 5)	2.446982	3.39458e-14	1.000000	87.089985	0	1.26205e-01
RAND(0, 10, 10)	5.824257	6.83614e-12	1.000000	455.743311	0	5.67927e-01
RAND(0, 10, 20)	9.183330	5.41627e-09	1.000000	2134.554441	1	3.25248e+00
RAND(0, 10, 30)	11.536522	1.93411e-08	1.000000	4984.086452	0	2.13387e+00
RAND(0, 10, 50)	15.243998	4.98166e-09	1.000000	14279.027031	13	7.13160e+01
RAND(0, 10, 100)	1.000011	6.04386e-11	1.000000	58632.497378	21	3.59539e+02
RAND(0, 100, 3)	2.633042	3.65720e-16	1.000000	119.520429	0	7.54934e-02
RAND(0, 100, 5)	10.565073	1.40079e-08	1.000000	837.420623	3	9.18979e+00
RAND(0, 100, 10)	1.000023	3.14679e-10	1.000000	4632.781263	11	1.44446e+01
RAND(0, 100, 20)	1.000033	6.10497e-10	1.000000	20711.816072	15	5.68845e+01
RAND(0, 100, 30)	1.000006	1.87741e-11	1.000000	48475.452360	19	7.35683e+01
RAND(0, 100, 50)	1.000046	1.10611e-09	1.000000	140942.375463	19	2.78063e+02
RAND(0, 100, 100)	1.000069	2.44184e-09	1.000000	590448.766570	23	1.05961e+03
RAND(0, 300, 3)	10.784435	1.27460e-13	1.000000	674.123199	0	3.88764e-01
RAND(0, 300, 5)	1.000018	2.49321e-10	1.000000	2284.241728	9	2.36784e+01
RAND(0, 300, 10)	1.000071	3.08010e-09	1.000000	13405.223524	13	4.14812e+01
RAND(0, 300, 20)	1.000004	9.08293e-12	1.000000	66894.917939	19	7.69437e+01
RAND(0, 300, 30)	1.000038	7.82981e-10	1.000000	150981.249208	19	1.31747e+02
RAND(0, 300, 50)	1.000048	1.20049e-09	1.000000	437408.346927	21	3.18451e+02
RAND(0, 300, 100)	1.000065	2.20512e-09	1.000000	1766076.567513	23	8.84416e+02

Table 9 Performance of the enumerative method for solving QEiCP$(B, 0, -C)$.

many applications lead to more general eigenvalue complementarity problems, where

	FB function				min function			
Problem	λ	Ntime	Nodes	CPU	λ	Ntime	Nodes	CPU
RAND(0, 1, 100)	6.911841	1	0	1.10600e+01	6.911841	1	0	1.12283e+01
RAND(0, 10, 20)	9.182325	1	0	1.21474e+00	9.182325	1	0	1.23527e+00
RAND(0, 10, 50)	15.241381	1	0	1.04002e+01	15.241381	1	0	1.03022e+01
RAND(0, 10, 100)	21.929891	1	0	4.19420e+01	21.929891	1	0	4.19704e+01
RAND(0, 100, 5)	10.564909	1	0	2.75402e+00	10.564909	1	0	2.83090e+00
RAND(0, 100, 10)	18.401286	1	0	2.72986e+00	18.401286	1	0	2.63761e+00
RAND(0, 100, 20)	29.002661	1	0	3.74956e+00	29.002661	1	0	3.88819e+00
RAND(0, 100, 30)	36.295914	1	0	6.48282e+00	36.295914	1	0	6.48330e+00
RAND(0, 100, 50)	47.957940	4	7	1.63330e+02	47.957940	4	7	1.63667e+02
RAND(0, 100, 100)	69.780913	2	3	2.28199e+02	69.780913	2	3	2.28554e+02
RAND(0, 300, 5)	20.232286	1	1	1.06610e+01	20.232286	1	1	1.07462e+01
RAND(0, 300, 10)	31.862852	1	0	3.25188e+00	31.862852	1	0	3.28776e+00
RAND(0, 300, 20)	52.304464	1	1	1.65669e+01	52.304464	1	1	1.64167e+01
RAND(0, 300, 30)	64.155291	1	0	1.69281e+01	64.155291	1	0	1.74003e+01
RAND(0, 300, 50)	84.828259	5	9	2.09547e+02	84.828259	5	9	2.09481e+02
RAND(0, 300, 100)	120.743763	3	5	3.35190e+02	120.743763	3	5	3.42279e+02

Table 10 Performance of the hybrid method for solving QEiCP($B, 0, -C$).

the investigation of such approaches seems to be worthwhile to pursue in future studies.

References

1. Adly, S., Seeger, A.: A nonsmooth algorithm for cone constrained eigenvalue problems. Computational Optimization and Applications 49, 299-318 (2011)

2. 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 8, 197-215 (2012)

3. Brás, C., Iusem, A.N., Júdice, J.: On the quadratic eigenvalue complementarity problem (submitted).

4. Brás, C., Fukushima, M., Iusem, A.N., Júdice, J.: On the conic eigenvalue complementarity problem (submitted).

5. Brooke, A., Kendrik, D., Meeraus, A., Raman, R.: GAMS: A Users' Guide. GAMS Development Corporation, Washington (1998)

6. Cottle, R.W., Pang, J.S., Stone, R.S.: The Linear Complementarity Problem. Academic Press, New York (1992)

7. Facchinei, F., Pang, J.S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, Berlin (2003)

8. Fernandes, L. M., Júdice, J., Fukushima, M., Iusem, A.: On the symmetric quadratic eigenvalue complementarity problem. Optimization Methods and Software 29, 751-770 (2014)

9. Fernandes, L. M., Júdice, J., Sherali, H. D., Forjaz, M.A.: On an enumerative algorithm for solving eigenvalue complementarity problems. Computational Optimization and Applications 59, 113-134 (2014)

10. Fernandes, L.M., Júdice, J., Sherali, H. D., Fukushima, M.: On the computation of all eigenvalues for the eigenvalue complementarity problems. Journal of Global Optimization 59, 307-326 (2014)

11. Júdice, J., Raydan, M., Rosa, S., Santos, S.: On the solution of the symmetric complementarity problem by the spectral projected gradient method. Numerical Algorithms 44, 391-407 (2008)

12. Júdice, J, Sherali, H. D., Ribeiro, I.: The eigenvalue complementarity problem. Computational Optimization and Applications 37, 139-156 (2007)

13. Júdice, J., Sherali, H. D., Ribeiro, I., Rosa, S.: On the asymmetric eigenvalue complementarity problem. Optimization Methods and Software 24, 549-586 (2009)

14. Le Thi, H., Moeini, M., Pham Dinh, T., Júdice, J.: A DC programming approach for solving the symmetric eigenvalue complementarity problem. Computational Optimization and Applications 5, 1097-1117 (2012)

15. Murty, K.: Linear Complementarity, Linear and Nonlinear Programming. Heldermann, Berlin (1988)

16. MATLAB, version 8.0.0.783 (R2012b). The MathWorks Inc., Natick, MS (2012)

17. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (2006)

18. Niu, Y.S., Pham, T. Le Thi, H.A., Júdice, J.: Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem. Optimization Methods and Software 28, 812-829 (2013)

19. Pinto da Costa, A., Martins, J., Figueiredo, I., Júdice, J.: The directional instability problem in systems with frictional contact. Computer Methods in Applied Mechanics and Engineering 193, 357-384 (2004)

20. Pinto da Costa, A., Seeger, A.: Cone constrained eigenvalue problems, theory and algorithms. Computational Optimization and Applications 45, 25-57 (2010)

21. Queiroz, M., Júdice, J., Humes, C.: The symmetric eigenvalue complementarity problem. Mathematics of Computation 73, 1849-1863 (2003)

22. Seeger, A.: Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions. Linear Algebra and Its Applications 294, 1-14 (1999)

23. Seeger, A.: Quadratic eigenvalue problems under conic constraints. SIAM Journal on Matrix Analysis and Applications 32, 700-721 (2011)

24. Seeger, A., Torky, M.: On eigenvalues induced by a cone constraint. Linear Algebra and Its Applications 372, 181-206 (2003)

25. Sherali, H.D., Adams, W.P.: A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Kluwer, Dordrecht (1999)

26. Tawarmalani, M. Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Mathematical Programming 103, 225-249 (2005)

27. Wächter, A., Biegler, L.T.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25-57 (2006)