On the Computation of All Eigenvalues for the Eigenvalue Complementarity Problem

Luís M. Fernandes* Joaquim J. Júdice† Hanif D. Sherali‡ Masao Fukushima§

November 9, 2012

Abstract

In this paper, a parametric algorithm is introduced for computing all eigenvalues for two Eigenvalue Complementarity Problems discussed in the literature. The algorithm searches a finite number of nested intervals $[\bar{l}, \bar{u}]$ in such a way that, in each iteration, either an eigenvalue is computed in $[\bar{l}, \bar{u}]$ or a certificate of nonexistence of an eigenvalue in $[\bar{l}, \bar{u}]$ is provided. A hybrid method that combines an enumerative method [5] and a semi-smooth algorithm [1] is discussed for dealing with the Eigenvalue Complementarity Problem over an interval $[\bar{l}, \bar{u}]$. Computational experience is presented to illustrate the efficacy and efficiency of the proposed techniques.

Keywords: Eigenvalue Problems, Complementarity Problems, Nonlinear Programming, Global Optimization.

Mathematics Subject Classification: 90B60, 90C33, 90C30, 90C26

1 Introduction

The *Eigenvalue Complementarity Problem* (**EiCP**) [11, 15] involves finding a real number λ and a vector $x \in \mathbb{R}^n$ such that

$$w = (\lambda B - A)x \tag{1}$$

$$w \geq 0, x \geq 0 \tag{2}$$

$$x^T w = 0 \tag{3}$$

$$e^T x = 1, \tag{4}$$

where $e \in \mathbb{R}^n$ is a vector of ones, $w \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ are given matrices, and B is positive definite (PD), i.e., $x^T Bx > 0$ for all $x \neq 0$. The real number λ is called a *complementary eigenvalue* and the corresponding vector x a *complementary eigenvector*. This problem finds many interesting applications in different areas of science and engineering [11, 17, 20]. The EiCP always

*Instituto Politécnico de Tomar and Instituto de Telecomunicações, Portugal (lmerca@co.it.pt).

†Instituto de Telecomunicações, Portugal (joaquim.judice@co.it.pt).

‡Grado Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA, USA (hanifs@vt.edu) (This research is supported in part by the *National Science Foundation*, under Grant Number CMMI - 0969169).

§Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (fuku@i.kyoto-u.ac.jp) (The research of this author was supported in part by a Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science).

has a solution since it is equivalent to the following Variational Inequality Problem VI(F, Δ) [7]: Find a vector $x \in \Delta$ such that

$$F(x)^T(y-x) \geq 0, \ \forall y \in \Delta,$$

where $F\colon \mathbb{R}^n \backslash \{0\} \to \mathbb{R}^n$ is defined by

$$F(x) = \left(\frac{x^T A x}{x^T B x} B - A \right) x, \tag{5}$$

and Δ is the unit simplex:

$$\Delta = \{x \in \mathbb{R}^n \colon e^T x = 1, x \geq 0\}. \tag{6}$$

The *Quadratic Eigenvalue Complementarity Problem* (**QEiCP**) is an interesting extension of the EiCP that has recently been introduced in [16]. This problem has a number of important applications [16] and consists of finding $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n$ such that

$$w = \lambda^2 A x + \lambda B x + C x \tag{7}$$

$$w \geq 0, x \geq 0 \tag{8}$$

$$x^T w = 0 \tag{9}$$

$$e^T x = 1. \tag{10}$$

Contrary to the EiCP, the QEiCP may have no solution when the matrix A of the leading term is PD. Certain co-regular and co-hyperbolic properties were introduced in [16] as sufficient conditions for the QEiCP to have a solution. As for the EiCP, the QEiCP is equivalent in this case to VI(G, Δ), where $G\colon \mathbb{R}^n \to \mathbb{R}^n$ is an appropriate mapping and Δ is the unit simplex (6) [16].

A number of algorithms have been proposed during the past several years for finding an eigenvalue for the EiCP [1, 3, 6, 7, 8, 9, 12, 13, 17, 18]. Among these, a semi-smooth method [1] and an enumerative method [7] have proven to be most effective. These two algorithms have been recently extended to deal with the QEiCP when the co-regular and co-hyperbolic properties hold [5, 16].

In this paper, we address the problem of finding all the eigenvalues for the EiCP, and also for the QEiCP, under the co-regular and co-hyperbolic properties. A parametric algorithm is developed for this purpose that is able to find all the $\epsilon-$distinct eigenvalues. Note that, as formalized in the sequel, λ_1 and λ_2 are said to be $\epsilon-$distinct if $|\lambda_1 - \lambda_2| > 2\epsilon$. This parametric algorithm relies on the solution of a finite number of EiCPs or QEiCPs pertaining to λ belonging to certain specified intervals $[\bar{l}, \bar{u}]$. In order to solve these EiCP or QEiCPs, a modified version of the enumerative method [7] discussed in [5] is utilized. This procedure either finds an eigenvalue belonging to $[\bar{l}, \bar{u}]$ or shows that no eigenvalue exists in this interval. A hybrid algorithm that combines the beneficial features of both the enumerative method and of the semi-smooth method [1] is also introduced for dealing with the EiCP or QEiCP over the defined intervals $[\bar{l}, \bar{u}]$.

Computational experience using a variety of EiCPs and QEiCPs of dimension up to $n = 50$ reveals that the parametric algorithm is efficient for computing all the $\epsilon-$distinct eigenvalues for the EiCP and QEiCP. Furthermore, the hybrid method turns out to be effective for solving the EiCP or QEiCP associated with the first interval generated by the parametric algorithm, but then the relatively more simple version of the enumerative method is recommended to be used from then on.

The structure of the remainder of this paper is as follows. In Section 2, the problem of finding all the eigenvalues for the EiCP or QEiCP is addressed together with the parametric algorithm. The enumerative, semi-smooth and hybrid methods are discussed in Section 3. Computational experiments with these algorithms and some concluding remarks are presented in the last section of the paper.

2 Finding all the eigenvalues for EiCP and QEiCP

In this section, we start by addressing the problem of computing all the eigenvalues for the EiCP, i.e., all the real numbers $\bar{\lambda}$ for which there exist vectors $x \in \mathbb{R}^n$ and $w \in \mathbb{R}^n$ satisfying the conditions (1)-(4) with $\lambda = \bar{\lambda}$. Traditional (or generalized) eigenvalues for the Unconstrained Eigenvalue Problem (B,A) may help for this purpose, as the following result trivially holds:

Theorem 1. *If $\bar{\lambda}$ is an eigenvalue for the EiCP, then there is a set $I \subseteq \{1, \ldots, n\}$ such that $\bar{\lambda}$ is a generalized eigenvalue of (B_{II}, A_{II}), i.e., it satisfies*

$$\bar{\lambda} B_{II} x_I = A_{II} x_I \tag{11}$$

for an appropriate vector $x_I \in \mathbb{R}^{|I|} \backslash \{0\}$, where B_{II}, A_{II}, and x_I restrict the respective components of B, A, and x to the index set I.

Based on this result, it is possible, at least in theory, to compute all the eigenvalues for the EiCP by the following procedure:

All Principal Submatrices Algorithm

For each set $I \subseteq \{1, \ldots, n\}$ do:

(i) Compute all the real eigenvalues of (B_{II}, A_{II}).

(ii) For each real eigenvalue computed in Step (i), find a vector $x_I \in \mathbb{R}^{|I|}$ such that

$$\sum_{j \in I} x_j = 1 \tag{12}$$

$$x_j \geq 0, \qquad j \in I \tag{13}$$

$$\sum_{j \in I} (\bar{\lambda} b_{ij} - a_{ij}) x_j \geq 0, \qquad i \notin I. \tag{14}$$

(iii) If such a vector x_I exists, then add $\bar{\lambda}$ to the list of computed eigenvalues for the EiCP.

It is obvious that this procedure is finite but is not practical even for small values of n. For instance, for $n = 5$, it requires finding all the traditional eigenvalues for $2^5 - 1 = 31$ Eigenvalue Problems, and then solving a linear program for each of these real eigenvalues in order to verify whether the conditions (12)-(14) hold. However, this type of procedure may be efficient for some special cases, as the one presented below.

Theorem 2. *If $A > 0$ and $B = I$, then EiCP (1)-(4) has a unique eigenvalue.*

Proof. Since $A > 0$, then by [2], there is a $\lambda > 0$ and a vector x satisfying $x > 0$, $e^T x = 1$, and

$$Ax = \lambda x.$$

Thus, (λ, x) is a solution of EiCP. Since λ is a traditional eigenvalue, then it is also an eigenvalue of $A^T > 0$, and there exists another vector y such that $y > 0$ and $e^T y = 1$ such that

$$A^T y = \lambda y. \tag{15}$$

Now, suppose that $\bar{\lambda} \neq \lambda$ is another solution of EiCP (1)-(4). Since $A > 0$, then it immediately follows that $\bar{\lambda}$ must be a traditional eigenvalue of A, so that there exists a vector $\bar{x}$ such that $\bar{x} \geq 0$, $e^T\bar{x} = 1$, and

$$A\bar{x} = \bar{\lambda}\bar{x}. \tag{16}$$

Hence, by (15) and (16), we have

$$\begin{aligned} y^T A\bar{x} &= \bar{\lambda} y^T \bar{x} \\ y^T A\bar{x} &= \bar{x}^T(A^T y) = \lambda y^T \bar{x}. \end{aligned}$$

These equations imply that

$$(\bar{\lambda} - \lambda) y^T \bar{x} = 0.$$

Since $y > 0$ and $0 \neq \bar{x} \geq 0$, then $\bar{\lambda} = \lambda$, a contradiction. □

Note that this condition is not necessary for $B = I$, as the EiCP with $B = I$ and $A = 0$ also has a unique eigenvalue.

The previous discussion indicates that the problem of finding all the eigenvalues in general requires a completely different approach. In [5], we derived a simple procedure for computing an interval that contains all the eigenvalues for the EiCP. The main idea of the algorithm proposed in the present paper for computing all the complementary eigenvalues is to progressively partition this interval into subintervals that contain complementary eigenvalues in a systematic way. In order to explain the algorithm, let $[l, u]$ be the interval that contains all the complementary eigenvalues. By applying the enumerative method discussed in [5] over this interval, a complementary eigenvalue $\tilde{\lambda} \in [l, u]$ is computed. In the next iteration, two intervals $[l, \tilde{\lambda} - \epsilon]$ and $[\tilde{\lambda} + \epsilon, u]$ are constructed and the enumerative method is applied for λ belonging to each of these intervals. If $[\bar{l}, \bar{u}]$ represents a currently considered interval for which the enumerative method is applied, then two cases may occur:

(i) There is no complementary eigenvalue in $[\bar{l}, \bar{u}]$, and so this interval is discarded from further investigation.

(ii) A new complementary eigenvalue $\bar{\lambda} \in [\bar{l}, \bar{u}]$ is computed, and therefore two new subintervals $[\bar{l}, \bar{\lambda} - \epsilon]$ and $[\bar{\lambda} + \epsilon, \bar{u}]$ are added for further investigation.

Each EiCP associated with an interval $[\bar{l}, \bar{u}]$ may be solved by using the enumerative method discussed in [5], which is applied to the following linearly constrained nonlinear program in an expanded (x, y, w, λ)-space:

$$\begin{array}{lll} \textbf{NLP:} & \text{Minimize} & \|y - \lambda x\|_2^2 + x^T w = f(x, y, w, \lambda) \\ & \text{subject to} & w = By - Ax \\ & & e^T x = 1 \\ & & e^T y = \lambda \\ & & \bar{l} x_j \leq y_j \leq \bar{u} x_j, \forall j \\ & & w \geq 0, x \geq 0 \\ & & \bar{l} \leq \lambda \leq \bar{u}. \end{array} \tag{17}$$

Note that $\bar{l}$ (respectively, $\bar{u}$) is equal to l (respectively, u) or $\bar{\lambda} + \epsilon$ (respectively, $\bar{\lambda} - \epsilon$), where $\bar{\lambda}$ is a previously computed eigenvalue. The algorithm either terminates with a solution (within a prescribed

tolerance) or with a certificate that no solution exists (list of open nodes is empty). Note that a node k corresponding to an interval $[\bar{l}\,, \bar{u}]$ should be fathomed if $\bar{u} - \bar{l} < \epsilon$ or if the corresponding NLP (17) is infeasible.

The procedure is repeated until there exists no interval left to be investigated. The formal steps of this algorithm are presented below.

Parametric algorithm for computing all complementary eigenvalues

Step 0 (Initialization) - Let ϵ be a positive tolerance, let $\bar{l} = l$ and $\bar{u} = u$ be computed by using the procedures discussed in [5], where $[l\,, u]$ captures all the eigenvalues of the EiCP, and let $L = \{1\}$ be the set of open nodes.

Step 1 (Choice of node) - If $L = \emptyset$ terminate. Otherwise select $k \in L$ with the corresponding interval $[\bar{l}, \bar{u}]$, and let EiCP denote the associated problem.

Step 2 (Solve and Branch) - If $\bar{u} - \bar{l} \leq \epsilon$ or NLP (17) is infeasible, then disregard $[\bar{l}, \bar{u}]$; replace $L \leftarrow L\backslash\{k\}$ and return to Step 1. Otherwise, let $(x^*, y^*, w^*, \lambda^*)$ be a global minimum of NLP (17). If $f(x^*, y^*, w^*, \lambda^*) > \epsilon$, set $L \leftarrow L\backslash\{k\}$ and return to Step 1. Otherwise, partition the interval $[\bar{l}, \bar{u}]$ at node k into $[\bar{l}, \lambda^* - \epsilon]$ and $[\lambda^* + \epsilon, \bar{u}]$ to generate two nodes (these respective nodes are not generated if $\lambda^* < \bar{l} + \epsilon$ or $\lambda^* > \bar{u} - \epsilon$).

Step 3 (Queue) - Set $L \leftarrow L\backslash\{k\} \cup \{$ new nodes from Step 2 $\}$ and return to Step 1.

The following result establishes the finiteness of this algorithm.

Theorem 3. *Upon termination of the parametric algorithm, any missing eigenvalue lies within* 2ϵ *of a detected eigenvalue.*

Proof. By contradiction, suppose not. Then one of the following three cases must occur:

Case (i): There exists a missing eigenvalue λ^* for which $\lambda_i < \lambda^* < \lambda_{i+1}$, where $\lambda_i < \lambda_{i+1}$ are some two adjacent detected eigenvalues, and where by hypothesis, we have

$$\lambda^* - \lambda_i > 2\epsilon \quad \text{and} \quad \lambda_{i+1} - \lambda^* > 2\epsilon. \tag{18}$$

In this case, when λ_i was found, an interval was created to its right with a lower bound of $LB = \lambda_i + \epsilon$, and similarly, when λ_{i+1} was found, an interval was created to its left with an upper bound of $UB = \lambda_{i+1} - \epsilon$. Hence, in the open list of intervals, there existed an interval $[LB, UB]$ that should have been selected for scanning. But by (18), we have that $\lambda^* \in [LB, UB]$, where $(UB - LB) = \lambda_{i+1} - \lambda_i - 2\epsilon > (\lambda^* + 2\epsilon) + (2\epsilon - \lambda^*) - 2\epsilon = 2\epsilon$, which is a contradiction to the adjacency of λ_i and λ_{i+1} since $[LB, UB]$ should then have been scanned (since $UB - LB > \epsilon$) and some eigenvalue should have been found within this interval (since, in particular, $\lambda^* \in [LB, UB]$).

Case (ii): There exists a missing eigenvalue λ^* for which $l \leq \lambda^* < \lambda_1$, where λ_1 is the smallest detected eigenvalue, and where by hypothesis, we have

$$\lambda_1 - \lambda^* > 2\epsilon. \tag{19}$$

In this case, when the eigenvalue λ_1 was found, we created an interval to its left of the form $[l, \lambda_1 - \epsilon]$, where by (19), we have

$$\lambda^* \in [l, \lambda_1 - \epsilon], \quad \text{and} \quad (\lambda_1 - \epsilon) - l > (\lambda^* + 2\epsilon - \epsilon) - l \geq \epsilon.$$

This contradicts that λ_1 is the smallest detected eigenvalue since the interval $[l, \lambda_1 - \epsilon]$ should have been scanned by the algorithm because its length exceeds ϵ, and some eigenvalue within this interval should have been found because, in particular, λ^* belongs to this interval.

Case (iii): There exists a missing eigenvalue λ^* for which $\lambda_p < \lambda^* \leq u$, where λ_p is the largest detected eigenvalue, and where by hypothesis, we have $\lambda^* - \lambda_p > 2\epsilon$. This case leads to a similar contradiction as for Case (ii) that λ_p is the largest eigenvalue found by the algorithm. □

In [5], an enumerative method was developed for finding a global minimum of the NLP (17). Furthermore the number of NLPs (17) to be solved is finite. Hence, the parametric algorithm is able, in theory, to compute all the $\epsilon-$distinct eigenvalues for the EiCP. Note that, in light of Theorem 3, we consider two eigenvalues λ_1 and λ_2 to be $\epsilon-$distinct if and only if $|\lambda_1 - \lambda_2| > 2\epsilon$, where ϵ is the tolerance used in the parametric algorithm. The method can also be used for finding all the eigenvalues of the following Quadratic Eigenvalue Complementarity Problem (QEiCP) discussed in [16]:

$$\begin{aligned} w &= \lambda^2 Ax + \lambda Bx + Cx \\ w &\geq 0, x \geq 0 \\ e^T x &= 1 \\ x^T w &= 0, \end{aligned} \tag{20}$$

under the co-regular and co-hyperbolic conditions stated in [16]. A nonlinear program similar to NLP (17) needs to be processed at each iteration of the parametric algorithm, by using a direct extension of the enumerative method discussed in [5].

A drawback of the parametric algorithm is that some eigenvalues may be lost if they are too close to be considered as $\epsilon-$distinct. A possible way to avoid this phenomenon is not to use the tolerance ϵ in the parametric algorithm and to consider the following barrier problem instead of NLP (17):

$$\begin{aligned} \textbf{PNLP:} \quad \text{Minimize} \quad & \|y - \lambda x\|_2^2 + x^T w - \rho \log[(\lambda - \bar{l})(\bar{u} - \lambda)] \\ \text{subject to} \quad & w = By - Ax \\ & e^T x = 1 \\ & e^T y = \lambda \\ & \bar{l} \leq \lambda \leq \bar{u} \\ & w \geq 0, x \geq 0, \end{aligned} \tag{21}$$

along with a similar penalty problem for the QEiCP. In Section 4, we investigate whether this type of program can help the parametric method find all the eigenvalues for the EiCP and QEiCP.

3 A hybrid algorithm for computing an eigenvalue in a given interval

In this section, we discuss approaches for solving the EiCP for λ belonging to a particular interval $[\bar{l}, \bar{u}]$ or to show that no complementary eigenvalue exists in this interval. As stated before, this can be accomplished by finding a global minimum to NLP (17). An enumerative method was introduced in

[7] and subsequently improved and extended for QEiCP in [5]. This algorithm explores a binary tree that is constructed using two branching strategies, namely, based on a pair of positive complementary variables at the current stationary point of the Problem NLP (17) and by partitioning the interval $[\bar{l}, \bar{u}]$. Therefore, each node k of the resulting enumeration tree is associated with an interval $[\tilde{l}, \tilde{u}] \subseteq [\bar{l}, \bar{u}]$ along with two sets I and J that respectively record those $w-$ and $x-$variables that are presently fixed to zero. Since $y_i = \lambda x_i$, $i = 1, \ldots, n$, in any solution to the EiCP, the following constraints are thus associated with node k of this tree:

$$\begin{gathered} \bar{l} x_i \leq y_i \leq \bar{u} x_i, \quad \forall i \in \bar{J} \\ y_i = x_i = 0, \quad \forall i \in J \\ w_i = 0, \quad \forall i \in I, \end{gathered}$$

where $l \leq \bar{l} < \bar{u} \leq u$, $J \subseteq \{1, \ldots, n\}$, $\bar{J} = \{1, \ldots, n\} \backslash J$, and $J \cap I = \emptyset$. Furthermore, consider the sets

$$K = I \cup J, \ \bar{I} = \{1, \ldots, n\} \backslash I, \text{ and } \bar{K} = \{1, \ldots, n\} \backslash K.$$

Then the subproblem at node k is given as follows, where any set-subscript on a variable restricts the variable indices to the corresponding set:

$$\begin{array}{lll} \textbf{NLP(}k\textbf{):} & \text{Minimize} & f(x, y, w, \lambda) = (y_{\bar{J}} - \lambda x_{\bar{J}})^T (y_{\bar{J}} - \lambda x_{\bar{J}}) + x_{\bar{K}}^T w_{\bar{K}} \\ & \text{subject to} & w = By - Ax \\ & & e^T x_{\bar{J}} = 1 \\ & & e^T y_{\bar{J}} = \lambda \\ & & \bar{l} \leq \lambda \leq \bar{u} \\ & & \bar{l} x_j \leq y_j \leq \bar{u} x_j \,, \quad \forall j \in \bar{J} \\ & & w_{\bar{I}} \geq 0, x_{\bar{J}} \geq 0 \\ & & y_j = x_j = 0, \quad \forall j \in J \\ & & w_i = 0, \quad \forall i \in I. \end{array}$$

At this node k, the algorithm searches for a stationary point to the corresponding program NLP(k). If the objective function value at this stationary point is zero, then a solution to the EiCP is at hand and the algorithm terminates. Otherwise, two new nodes are created and the process is repeated. The algorithm includes heuristic rules for choosing an open node from some associated list and for deciding which of the two branching strategies should be used at the selected node k whenever a stationary point having a positive objective function value is found for NLP(k). The algorithm also gives a certificate of nonexistence of a complementary eigenvalue in the interval $[\bar{l}, \bar{u}]$ when it terminates with an empty list of nodes without having found a stationary point of NLP with an objective function value smaller than a positive tolerance. The formal steps of the algorithm are presented below.

Enumerative algorithm for finding a complementary eigenvalue in $[\bar{l}, \bar{u}]$.

Step 0 (Initialization) - Let ϵ_1 and ϵ_2 be selected tolerances, where $0 < \epsilon_1 < \epsilon_2$ (we can take $\epsilon_1 = \epsilon^2$ and $\epsilon_2 = \epsilon$ for some $0 < \epsilon < 1$, for example). Set $k = 1$, $[\tilde{l}, \tilde{u}] = [\bar{l}, \bar{u}]$, $I = \emptyset$, $J = \emptyset$, and find a stationary point $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$ of NLP(1). If NLP(1) is infeasible, then EiCP has no solution in $[\bar{l}, \bar{u}]$; terminate. Otherwise,

let $L = \{1\}$ be the set of open nodes, set $UB(1) = f(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$, and let $N = 1$ be the number of nodes generated.

Step 1 (Choice of node) - If $L = \emptyset$ terminate; EiCP has no solution in $[\bar{l}, \bar{u}]$. Otherwise, select $k \in L$ such that

$$UB(k) = \min\{UB(i) \colon i \in L\},$$

and let $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$ be the stationary point that was previously found at this node.

Step 2 (Branching rule) - Let

$$\theta_1 = \max\{\bar{w}_i \bar{x}_i \colon i \in \bar{K}\} = \bar{w}_r \bar{x}_r, \text{ and}$$
$$\theta_2 = \max\{|\bar{y}_i - \bar{\lambda}\bar{x}_i| \colon i \in \bar{J}\}\ .$$

(i) If $\theta_1 \leq \epsilon_1$ and $\theta_2 \leq \epsilon_2$, then $\bar{\lambda}$ yields a complementary eigenvalue (within the tolerance ϵ_2) with $\bar{x}$ being a corresponding eigenvector; terminate.
(ii) If $\theta_1 > \theta_2$, branch on the complementary variables (w_r, x_r) associated with θ_1 and generate two new nodes, $N + 1$ and $N + 2$, which respectively restrict $w_r = 0$ and $x_r = 0$.
(iii) If $\theta_1 \leq \theta_2$, then partition the interval $[\tilde{l}, \tilde{u}]$ at node k into $[\tilde{l}, \tilde{\lambda}]$ and $[\tilde{\lambda}, \bar{u}]$ to generate two new nodes, $N + 1$ and $N + 2$, where

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

Step 3 (Solve, Update, and Queue) - For each of $t = N + 1$ and $t = N + 2$ such that $\tilde{u} - \tilde{l} > \epsilon$, find a stationary point $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{\lambda})$ of Problem NLP(t). If NLP(t) is feasible, set $L = L \cup \{t\}$ and $UB(t) = f(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{\lambda})$. Set $L = L \backslash \{k\}$ and return to Step 1.

Reference [5] establishes the convergence of this algorithm and discusses its extension to the QEiCP.

Another interesting approach for computing a complementary eigenvalue for the EiCP is the *semi-smooth algorithm* [1], which is decribed as follows: Let the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ be defined by

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

This function is called the Fischer-Burmeister function and satisfies the following relations [4]:

$$\begin{aligned} x^T w = 0,\ x \geq 0,\ w \geq 0 \quad &\Leftrightarrow \quad x_i w_i = 0,\ x_i \geq 0,\ w_i \geq 0,\ \forall i = 1, \ldots, n \\ &\Leftrightarrow \quad \varphi(x_i, w_i) = 0,\ \forall i = 1, \ldots, n. \end{aligned}$$

Define the functions $\Phi : \mathbb{R}^{2n} \to \mathbb{R}^n$ and $\Psi : \mathbb{R}^{2n+1} \to \mathbb{R}^{2n+1}$ by

$$\Phi(x, w) = \begin{pmatrix} \varphi(x_1, w_1) \\ \vdots \\ \varphi(x_n, w_n) \end{pmatrix}$$

and

$$\Psi(x,w,\lambda)=\left(\begin{array}{c}\Phi(x,w)\\(\lambda B-A)x-w\\e^Tx-1\end{array}\right).$$

Then it is not difficult to see that the EiCP is equivalent to the system of equations

$$\Psi(x,w,\lambda)=0. \tag{22}$$

It should be noticed that the function φ is not differentiable at the origin, and so the system of equations (22) is nonsmooth. However it is well known [4] that the function Ψ enjoys a desirable property called semi-smoothness, which plays an important role in establishing superlinear convergence of the generalized (semi-smooth) Newton algorithm for solving nonsmooth equations. In order to explain the algorithm, let $\bar{z}=(\bar{x},\bar{w},\bar{\lambda})$ be a current point satisfying $e^T\bar{x}=1$. For simplicity of notation, let $\bar{y}=[\bar{y}_i]\in\mathbb{R}^n$ be the vector with components $\bar{y}_i=\sqrt{\bar{x}_i^2+\bar{w}_i^2}$. Since $e^T\bar{x}=1$, then $\bar{z}=(\bar{x},\bar{w},\bar{\lambda})$ is regarded as a solution of EiCP if and only if

$$\|\bar{w}-\bar{\lambda}B\bar{x}+A\bar{x}\|<\epsilon_1,\qquad \|\bar{y}-(\bar{x}+\bar{w})\|<\epsilon_2 \tag{23}$$

for sufficiently small ϵ_1, $\epsilon_2>0$. If (23) does not hold, then we update the current point by applying a Newton iteration. Specifically, the semi-smooth Newton direction $d=(d_x,d_w,d_\lambda)$ is obtained as a solution of the linear system

$$G(\bar{z})\left[\begin{array}{c}d_x\\d_w\\d_\lambda\end{array}\right]=\left[\begin{array}{c}\bar{y}-(\bar{x}+\bar{w})\\\bar{w}-\bar{\lambda}B\bar{x}+A\bar{x}\\0\end{array}\right], \tag{24}$$

where $G(\bar{z})$ is the Clarke generalized Jacobian of Ψ at $\bar{z}$, which can be computed as

$$G(\bar{z})=\left[\begin{array}{ccc}E&F&0\\\bar{\lambda}B-A&-I_n&B\bar{x}\\e^T&0&0\end{array}\right]\in\mathbb{R}^{(2n+1)\times(2n+1)}. \tag{25}$$

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

$$(E_{ii},F_{ii})=\left\{\begin{array}{ll}\left(1-\dfrac{\bar{x}_i}{\bar{y}_i},1-\dfrac{\bar{w}_i}{\bar{y}_i}\right) & \text{if } (\bar{x}_i,\bar{w}_i)\neq(0,0)\\(1-\xi_i,1-\eta_i) & \text{if } (\bar{x}_i,\bar{w}_i)=(0,0),\end{array}\right.\qquad \forall i=1,\ldots,n, \tag{26}$$

where (ξ_i,η_i) satisfies $\xi_i^2+\eta_i^2=1$. If $G(\bar{z})$ is nonsingular, then the direction (d_x,d_w,d_λ) is uniquely determined from (24) and a new point $\tilde{z}=(\tilde{x},\tilde{w},\tilde{\lambda})$ is obtained by

$$\tilde{x}=\bar{x}+d_x,\quad \tilde{w}=\bar{w}+d_w,\quad \tilde{\lambda}=\bar{\lambda}+d_\lambda, \tag{27}$$

which satisfies $e^T\tilde{x}=1$. A next iteration can now be performed with the new point $\tilde{z}$. The steps of the semi-smooth algorithm can be stated as follows:

Semi-Smooth Newton Algorithm

Step 0 (Initialization) - Let $(\bar{x}, \bar{\lambda}, \bar{w})$ be an initial point such that $e^T\bar{x} = 1$, and let ϵ_1 and ϵ_2 be selected positive tolerances.

Step 1 (Newton direction) - Compute the search direction by (24). If the matrix $G(\bar{z})$ given by (24) is singular, stop with an unsuccessful termination.

Step 2 (Update) - Find a new point $\tilde{z} = (\tilde{x}, \tilde{w}, \tilde{\lambda})$ by (27), and let $\bar{x} = \tilde{x}$, $\bar{w} = \tilde{w}$, and $\bar{\lambda} = \tilde{\lambda}$. If (23) holds, then stop with a solution $(\bar{x},\bar{\lambda})$ of EiCP. Otherwise, go to Step 1.

The above algorithm can be easily extended to deal with the QEiCP. In fact, only the expression of the Clarke Generalized Jacobian at $\bar{z}$ is different and takes the form

$$G(\bar{z}) = \begin{bmatrix} E & F & 0 \\ \bar{\lambda}^2 A + \bar{\lambda} B + C & -I_n & 2\bar{\lambda}A\bar{x} + B\bar{x} \\ e^T & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(2n+1)\times(2n+1)},$$

where as before, I_n is the identity matrix of order n, and E and F are the diagonal matrices whose diagonal elements are given by (26).

The semi-smooth algorithm has some benefits as well as drawbacks over the enumerative method. On the positive side, the semi-smooth method is in general fast in obtaining a solution for the EiCP or QEiCP whenever it is successful. Furthermore, each iteration of the algorithm does not require much effort and is simple to implement. However, on the negative side, the algorithm may terminate unsuccessfully with the singularity of the Generalized Jacobian or, even worse, may not be able to converge at all. The choice of the initial point is also very important for the algorithm to succeed in finding a complementary eigenvalue. Another drawback of the semi-smooth method is its inability to compute an eigenvalue in a particular interval $[\bar{l}, \bar{u}]$, as required by the parametric algorithm. In fact, this interval is not considered in the definition of the algorithm. It is possible to design special line-search techniques in the spirit of [4, 10] that guarantee a stationary point of an appropriate merit function in this interval. However, there is no guarantee that such a stationary point would provide a solution to the EiCP or to the QEiCP when it exists. Another negative feature of the semi-smooth method is its inability to provide a certificate of nonexistence of a complementary eigenvalue in a particular interval.

Recognizing all these drawbacks, but also benefits, of the semi-smooth algorithm, and since the enumerative method is able in theory to solve the EiCP or QEiCP in a particular interval $[\bar{l}, \bar{u}]$ but might be slow in achieving a complementary eigenvalue, we propose a *hybrid method* that tries to exploit the beneficial features of these two algorithms. Such a hybrid approach starts with the enumerative method. When the current $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$ is sufficiently close to a possible solution, i.e., when the value of the objective function of NLP (17) is sufficiently small, then the algorithm switches over to using the semi-smooth algorithm with the initial point $(\bar{x}, \bar{w}, \bar{\lambda})$. Then, either the algorithm finds a complementary eigenvalue within a maximum number of iterations, or it returns unsuccessfully back to the enumerative algorithm to continue with the previous point $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$. The switching decision is made using some additional tolerances $\bar{\epsilon_i} > \epsilon_i$, $i = 1, 2$, within the enumerative method. The formal steps of this hybrid method are presented below.

Hybrid Algorithm for finding a complementary eigenvalue in $[\bar{l}, \bar{u}]$.

Step 0 - Let $\bar{\epsilon}_1$ and $\bar{\epsilon}_2$ be two positive tolerances for switching from the enumerative algorithm to the semi-smooth method, and let $nmaxit$ be the maximum number of iterations permitted to be performed by the semi-smooth method.

Step 1 - Apply Step 1 of the enumerative method with positive tolerances ϵ_1 and ϵ_2 ($\epsilon_1 < \bar{\epsilon}_1$ and $\epsilon_2 < \bar{\epsilon}_2$). Let $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$ be the stationary point associated with the node k, and compute θ_1 and θ_2 as in Step 2 of the enumerative method.

(i) If $\theta_i < \epsilon_i$, $i = 1, 2$, stop with a solution of the EiCP.

(ii) If $\theta_i < \bar{\epsilon}_i$, $i = 1, 2$, go to Step 2.

(iii) Generate two new nodes as discussed in the enumerative method.

Step 2 - Apply the semi-smooth method. If the algorithm terminates with a solution (x^*, w^*, λ^*) such that $\lambda^* \in [\bar{l}, \bar{u}]$, stop. Otherwise the semi-smooth method terminates without success (singular generalized Jacobian, or number of iterations equal to $nmaxit$, or $\lambda^* \notin [\bar{l}, \bar{u}]$); go to Step 1 (iii) with node k and the solution $(\bar{x}, \bar{y}, \bar{w}, \bar{\lambda})$ given at the beginning of this step.

It is easy to see that this hybrid method retains the same convergence properties as the enumerative method, i.e., it either finds a solution of the EiCP or QEiCP in the interval $[\bar{l}, \bar{u}]$ or shows that there is no complementary $\epsilon-$distinct eigenvalue in this interval.

4 Computational Experiments

In this section, we report some computational experience with the parametric algorithm presented in the previous sections for the EiCP and QEiCP. Five sets of test problems were constructed for the EiCP, where B was always taken as the identity matrix. In the first set of test problems, the matrices A were taken from [1] and are given by

$$A = -\begin{bmatrix} 8 & -1 & 4 \\ 3 & 4 & 1/2 \\ 2 & -1/2 & 6 \end{bmatrix} \qquad \text{and} \qquad A = -\begin{bmatrix} 100 & 106 & -18 & -81 \\ 92 & 158 & -24 & -101 \\ 2 & 44 & 37 & -7 \\ 21 & 38 & 0 & 2 \end{bmatrix}.$$

These are denoted by SeegerAdly(n), where n is the order of the matrices ($n = 3$ and 4, respectively). For the second set of test problems [19], the matrix A is given by

$$A = -\begin{bmatrix} s^2 & s^3 & s^4 & s^5 & \dots \\ -s^3 & s^4 & s^5 & s^6 & \dots \\ -s^4 & s^5 & s^6 & s^7 & \dots \\ -s^5 & s^6 & s^7 & s^8 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

These test problems are denoted by SeegerVicente(n), where n is the order of the matrices, and where we used $s = \sqrt{6}$. In the third set of test problems [12], the matrix A is given by

$$A = - \begin{bmatrix} s^2 & s^3 & s^4 & s^5 & \dots \\ s^3 & s^4 & s^5 & s^6 & \dots \\ s^4 & s^5 & s^6 & s^7 & \dots \\ s^5 & s^6 & s^7 & s^8 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

These test problems are denoted by SeegerPCosta(n), where n is the order of the matrices, and where we used $s = 2$. In the fourth set of test problems, the matrix A was randomly generated with elements uniformly distributed in the intervals $[0, 1]$, $[-1, -1]$, and $[-10, 10]$. These problems are denoted by RAND(k, m, n), where k and m are the end-points of the chosen interval for generating the matrix elements, and n represents the order of the matrices A and B (we considered $n = 5, 10, 20, 30, 40$, and 50). In the last set of test problems, the matrix A is a block diagonal matrix and the elements of the diagonal blocks were randomly generated, uniformly distributed in the interval $[0, 1]$. These blocks have different dimensions and the problems are denoted by Block(k, m, s, n), where k and m are the end-points of the chosen interval for generating the matrix elements, s and n represent the number of blocks and the order of the matrices A and B, respectively (we considered $n = 5, 10, 20, 30, 40$, and 50). Since each diagonal block is a positive matrix, then by Theorem 2, the number of complementary eigenvalues is exactly equal to the number s of these diagonal blocks.

Two sets of test problems were considered for the QEiCP. The first test problem, denoted by AdlySeegerQ(3), has been taken from [1]. For all the remaining test problems, the matrix A was set equal to the identity matrix. Furthermore, the matrices B and $-C$ were randomly generated with their elements uniformly distributed in the interval $[0, 1]$ or $[0, 10]$, and these problems are denoted by RAND(k, m, n), where k and m are the end-points of the chosen interval for generating the matrix elements, and n represents the order of the matrices. In our experiments, the tolerances for the enumerative and hybrid methods were selected as $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 10^{-4}$, and $\bar{\epsilon}_1 = \bar{\epsilon}_2 = 10^{-1}$ (for different values of ϵ as specified below). Furthermore, we adopt the following notation:

- $n \equiv$ dimension of the EiCP or QEiCP, i.e., orders of their matrices.
- NEiv $\equiv$ number of eigenvalues ("$t?$" means that we actually do not know the exact number of complementary eigenvalues, but its number seems to be t; note that this uncertainty is due to the fact that we are only assured of finding a set of $\epsilon-$distinct eigenvalues).
- CEiv $\equiv$ number of computed eigenvalues by the parametric algorithm.
- NP $\equiv$ number of EiCPs or QEiCPs solved in nested sub-intervals of $[l, u]$ generated in the course of finding all the eigenvalues.
- l, $u \equiv$ end-points of the interval containing all the complementary eigenvalues.
- Nodes $\equiv$ total number of nodes generated.
- $\epsilon \equiv$ value of the tolerance used in the parametric algorithm.

Table 1 reports the numerical results obtained for our first experiment using the parametric algorithm with two different values of the tolerance ϵ. The simple version of the enumerative method (without the hybridization with the semi-smooth algorithm) has been used to process all the EiCPs in

Table 1: Performance of the parametric algorithm for solving EiCPs.

					$\epsilon = 10^{-4}$			$\epsilon = 10^{-3}$		
Problem	n	l	u	NEiv	CEiv	Nodes	NP	CEiv	Nodes	NP
SeegerAdly(3)	3	-13.000	1.718	9	21	3059	34	9	2750	19
SeegerAdly(4)	4	-346.000	224.157	23	66	9436	100	27	12228	51
RAND(0,1,5)	5	1.724	3.475	1	1	15	3	1	15	3
RAND(0,1,10)	10	2.779	5.906	1	1	25	3	1	25	3
RAND(0,1,20)	20	6.601	11.756	1	1	45	3	1	45	3
RAND(0,1,30)	30	12.751	18.389	1	1	65	3	1	65	3
RAND(0,1,40)	40	16.360	23.839	1	1	167	3	1	85	3
RAND(0,1,50)	50	18.418	29.756	1	1	207	3	1	105	3
RAND(-1,1,5)	5	-0.884	3.504	1?	1	25	3	1	25	3
RAND(-1,1,10)	10	-2.914	5.399	4?	9	2281	16	4	2077	9
RAND(-1,1,20)	20	-3.684	12.661	5?	39	8577	52	5	6039	11
RAND(-10,10,5)	5	-19.796	27.930	3?	5	205	9	3	177	7
RAND(-10,10,10)	10	-47.389	67.795	1?	2	825	4	1	780	3
RAND(-10,10,20)	20	-33.749	121.968	1?	4	1768	7	1	1511	3
SeegerVicente(3)	3	-340.182	59.0234	9	29	2808	47	9	2520	19
SeegerVicente(4)	4	-2129.271	165.8136	21	41	6885	71	21	4806	41
SeegerVicente(5)	5	-12991.629	441.8821	45	64	11762	100	39	9505	76
SeegerPCosta(3)	3	-112.000	0.0000	7	7	81	15	7	81	15
SeegerPCosta(4)	4	-480.000	0.0000	15	15	677	31	15	222	31
SeegerPCosta(5)	5	-1984.000	0.0000	31	31	2497	63	31	576	63

the intervals $[\bar{l}, \bar{u}]$ generated by the parametric algorithm. These results lead to the following conclusions:

- For $\epsilon = 10^{-4}$ the algorithm repeats the computation of some eigenvalues for a few test problems. This repetition only occurred in one test problem when $\epsilon = 10^{-3}$;
- $\epsilon = 10^{-3}$ seems to be the best choice for these test problems;
- The algorithm may be unable to compute some eigenvalues when they are quite close (non-$\epsilon-$distinct) due to the updating rule for the intervals. For instance, for Problem SeegerVicente(5), the algorithm with $\epsilon = 10^{-3}$ computes the eigenvalues $\lambda_i = 12.007767$ and $\lambda_{i+1} = 12.009029$, yielding an interval

$$[\lambda_i + \epsilon, \lambda_{i+1} - \epsilon] = [12.008029, 12.008767] = [\bar{l}, \bar{u}].$$

  Since $\bar{u} - \bar{l} < \epsilon = 10^{-3}$, the algorithm does not investigate this interval and the correct eigenvalues 12.007920, 12.0079522, and 12.008988 are lost. Note that, according to our definition of $\epsilon-$distinct solutions, these eigenvalues are considered to be non-$\epsilon-$distinct from one of the computed eigenvalues λ_i or λ_{i+1}. Furthermore, all the missing eigenvalues do indeed satisfy Theorem 3.

As a final conclusion of this experiment, the parametric algorithm seems to be efficient for finding all the distinct eigenvalues under a tolerance ϵ. This tolerance is necessary to avoid the repetition of eigenvalues, but its choice is the main drawback of the algorithm. In fact, if ϵ is too small, the algorithm tends to repeat the computation of eigenvalues. If ϵ is relatively large, then some close-by eigenvalues may be lost. It appears that $\epsilon = 10^{-3}$ offers a suitable compromise, and is used in our further experiments below with the parametric algorithm.

Table 2: Performance of the parametric algorithm for solving EiCPs and QEiCPs.

					$\epsilon = 10^{-3}$		
Problem	n	l	u	NEiv	CEiv	Nodes	NP
Block(0,1,2,5)	5	1.186	3.105	2	2	35	5
Block(0,1,4,10)	10	0.490	3.709	4	4	406	9
Block(0,1,8,20)	20	0.276	4.573	8	8	4225	16
Block(0,1,12,30)	30	0.500	5.667	12	12	10021	24
Block(0,1,16,40)	40	0.072	7.705	16	16	14889	32
Block(0,1,20,50)	50	0.114	6.639	20	20	20278	41
SeegerAdlyQ(3)	3	−10.875	5.469	12	12	179	25
RAND(0,1,5)	5	−4.944	2.669	4?	4	75	9
RAND(0,1,10)	10	−9.345	4.903	2?	2	415	5
RAND(0,1,20)	20	−19.596	10.042	2?	2	1645	5
RAND(0,1,30)	30	−29.585	15.037	4?	4	6079	9
RAND(0,1,40)	40	−39.555	20.022	2?	2	3100	5
RAND(0,10,5)	5	−42.789	21.607	2?	2	57	5
RAND(0,10,10)	10	−95.230	47.858	4?	4	424	9

Table 2 reports the performance of the parametric algorithm for computing all the eigenvalues for EiCPs having the defined block diagonal matrices and for the corresponding QEiCPs while using the tolerance $\epsilon = 10^{-3}$.

These results confirm the conclusions stated before regarding the efficacy of the parametric method. The algorithm was indeed able to find all the complementary eigenvalues for all the test problems having known eigenvalues. The enumerative method is, in general, efficient in finding an eigenvalue within a particular interval $[\bar{l}, \bar{u}]$ whenever it exists, but finds it more difficult to provide a certificate of nonexistence of a complementary eigenvalue otherwise. In order to reduce the total computational effort required by the parametric method, we have used a maximum number of nodes to be visited by the enumerative method when applied to an interval $[\bar{l}, \bar{u}]$. This number was chosen to be equal to 1500 but could be smaller. In all our experiments, this number was only achieved for a few Block test problems. Furthermore, when such a number of 1500 nodes was attained for a particular interval $[\bar{l}, \bar{u}]$, then no complementary eigenvalue existed in this interval, which is confirmed by the fact that the parametric method was indeed able to find all the complementary eigenvalues.

Table 3 presents the numerical results for the parametric algorithm when we solve the barrier penalty problem PNLP (21) instead of NLP (17) at each iteration. These PNLPs were solved by BARON [14] (with default options). We set $\rho = 10^{-4} < \epsilon$, since we wish $\lim_{\rho \to 0} \rho \log(\epsilon) = 0$. We considered a global minimum of PNLP to be an eigenvalue for EiCP if and only if

$$\min\{\bar{\lambda} - \bar{l}, \bar{u} - \bar{\lambda}\} \geq 10^{-4}, \text{ and } \theta_1 \leq 10^{-6}, \text{ and } \theta_2 \leq 10^{-4},$$

where θ_1 and θ_2 are the tolerances used in the stopping criterion of the enumerative method.

Problem PNLP (21) can also be solved by the enumerative algorithm and its convergence would follow exactly as in [5], where the stationary point problem for each node includes the convex barrier function term in the objective function with the original $(\bar{l}, \bar{u})-$values, while the algorithm branches on $(x_j = 0) \ \vee \ (w_j = 0)$ as well as on the $\lambda-$interval within the constraints.

Table 3: Performance of the parametric algorithm with the barrier problem and $\rho = 10^{-4}$.

			enumerative			BARON	
Problem	n	NEiv	CEiv	Nodes	NP	CEiv	NP
SeegerAdly(3)	3	9	3	39	7	8	17
SeegerAdly(4)	4	23	1	4	3	17	35
RAND(0,1,5)	5	1	4	70	9	1	3
RAND(0,1,10)	10	1	3	201	7	1	3
RAND(0,1,20)	20	1	1	631	3	1	3
RAND(0,1,30)	30	1	2	7	5	1	3
RAND(0,1,40)	40	1	1	89	3	1	3
RAND(0,1,50)	50	1	1	755	3	1	3
RAND(-1,1,5)	5	1?	1	4	3	1	3
RAND(-1,1,10)	10	4?	1	4	3	2	5
RAND(-1,1,20)	20	3?	2	794	5	1	3
RAND(-10,10,5)	5	3?	2	116	5	3	7
RAND(-10,10,10)	10	1?	1	7	3	1	3
RAND(-10,10,20)	20	1?	1	197	3	1	3
SeegerVicente(3)	3	9	2	10	5	8	17
SeegerVicente(4)	4	21	3	19	7	16	33
SeegerVicente(5)	5	45	5	64	11	33	67
SeegerPCosta(3)	3	7	2	7	5	7	15
SeegerPCosta(4)	4	15	1	4	3	15	31
SeegerPCosta(5)	5	31	3	21	7	31	63

When the only existing eigenvalue in $[\bar{l}, \bar{u}]$ is close to one of the end-points of this interval, the algorithm might take a long time to find this eigenvalue because of the barrier term. So, we further modified the algorithm as follows: Let $[\tilde{l}, \tilde{u}]$ be the current node's bounding interval. If $\bar{l} < \tilde{l} < \tilde{u} < \bar{u}$, then we find a stationary point without the barrier term. Else, suppose that $\tilde{l} = \bar{l}$ or $\tilde{u} = \bar{u}$. If $\tilde{u} - \tilde{l} < \epsilon$, we check whether we have an $\epsilon-$tolerance EiCP solution at hand, and we fathom the node. Otherwise, we solve the stationary point problem with the barrier term. Despite this modification, the numerical results displayed in Table 3 show that the parametric method based on PNLP is unable to compute all the eigenvalues, and more (though not all) eigenvalues were computed when BARON was used instead.

As a final conclusion of these experiments (and also as observed with many runs not reported in this table), we recommend using the proposed parametric method with the $\epsilon-$tolerance given by 10^{-3} for finding all the eigenvalues for the EiCP and QEiCP. The algorithm is, in general, efficient for finding $\epsilon-$distinct eigenvalues.

The second objective of our computational experiments was to investigate whether the hybrid method introduced in Section 3 offers an improvement over the simple version of the enumerative method for solving the EiCP or QEiCP in an interval generated by the parametric algorithm. In order to answer this question, we report some preliminary computational experience with this hybrid method for the test problems described in [5] for which the enumerative method was unable to find a solution of the EiCP at the root node. In our experiments, we used $\bar{\epsilon}_1 = 10^{-1}$ and $\bar{\epsilon}_2 = 10^{-1}$ and

$nmaxit = 100$. Table 4 presents the numerical results obtained. The following tolerances have been used:

- Tolerances for enumerative method: $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-4}$.
- Tolerances for Semi-Smooth Newton Algorithm: $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-6}$.

Furthermore, we adopt the following notation:

- $n \equiv$ dimension for the EiCP.
- Eigenvalue $\equiv$ eigenvalue computed by the algorithm.
- ITpivot $\equiv$ total number of pivotal operations.
- Nodes $\equiv$ total number of nodes generated.
- NTime $\equiv$ number of times that the Semi-Smooth Newton Algorithm is called.
- Iterations $\equiv$ number of iterations for the Semi-Smooth Newton Algorithm.
- $* \equiv$ the Semi-Smooth Newton Algorithm could not satisfy the Stopping Criterium (23) within 100 iterations.

Table 4: Performance of the enumerative algorithm vs. the hybrid algorithm for solving EiCPs.

		Enumerative algorithm			Hybrid Algorithm $\theta_1, \theta_2 \leq 10^{-1}$				
Problem	n	Eigenvalue	ITpivot	Nodes	Eigenvalue	ITpivot	Nodes	NTime	Iterations
SeegerAdly(3)	3	-4.134	19	0	-4.134	19	0	0	0
SeegerAdly(4)	4	-29.134	36	0	-29.134	36	0	0	0
SeegerVicente(3)	3	-24.000	14	0	-24.000	14	0	0	0
SeegerVicente(4)	4	-18.000	20	0	-18.000	20	0	0	0
SeegerVicente(5)	5	-12.010	27	0	-12.010	27	0	0	0
SeegerPCosta(3)	3	-4.000	5	0	-4.000	5	0	0	0
SeegerPCosta(4)	4	-4.000	5	0	-4.000	5	0	0	0
SeegerPCosta(5)	5	-4.000	2	0	-4.000	2	0	0	0
SeegerPCosta(10)	10	-4.000	11	0	-4.000	11	0	0	0
SeegerPCosta(20)	20	-4.000	11	0	-4.000	11	0	0	0
RAND(0,1,5)	5	2.781	15	0	2.781	15	0	0	0
RAND(0,1,10)	10	4.816	35	0	4.816	35	0	0	0
RAND(0,1,20)	20	9.850	66	0	9.850	66	0	0	0
RAND(0,1,30)	30	15.270	94	0	15.270	94	0	0	0
RAND(0,1,40)	40	20.241	131	0	20.241	131	0	0	0
RAND(0,1,50)	50	25.0772	166	0	25.0772	166	0	0	0
RAND(0,1,100)	100	49.755	381	0	49.755	381	0	0	0
RAND(-1,1,5)	5	1.123	24	0	1.123	24	0	0	0
RAND(-1,1,10)	10	-0.018	41	0	-0.018	41	0	0	0
RAND(-1,1,20)	20	0.842	105	0	0.842	105	0	0	0
RAND(-1,1,30)	30	2.346	1235	9	2.346	262	0	1	22
RAND(-1,1,40)	40	2.861	3107	17	2.865	392	0	1	16
RAND(-1,1,50)	50	3.131	3239	9	3.130	780	0	1	60
RAND(-1,1,100)	100	4.010	5500	6	3.948	1449	0	1	78
RAND(-10,10,5)	5	-9.922	29	2	-9.922	29	2	0	0
RAND(-10,10,10)	10	17.272	113	0	17.272	113	0	0	0
RAND(-10,10,20)	20	21.191	554	6	21.191	277	3	1	65
RAND(-10,10,30)	30	25.332	1562	6	25.332	600	0	1	69
RAND(-10,10,40)	40	19.595	3130	11	29.906	475	0	1	56
RAND(-10,10,50)	50	20.457	3160	6	20.457	3160	6	4	$4 \times 100^*$
RAND(-100,100,5)	5	135.146	23	0	135.146	23	0	0	0
RAND(-100,100,10)	10	-40.854	88	0	-40.854	88	0	0	0
RAND(-100,100,20)	20	80.072	378	0	80.072	378	0	0	0
RAND(-100,100,30)	30	180.221	1812	5	180.221	715	2	2	$100^* + 89$
RAND(-100,100,40)	40	234.283	2293	3	234.283	2293	3	0	0
RAND(-100,100,50)	50	176.117	4650	4	176.117	4650	4	1	100^*
RAND(-100,100,100)	100	526.975	7246	2	526.975	7246	2	0	0

The following observations can be made from the results in Table 4:

(i) For eight test problems, the semi-smooth algorithm was called once and successfully found a solution to the EiCP. In these examples, the hybrid method required fewer nodes and less number of pivotal operations than the enumerative method.

(ii) For two problems, the switching criterion $\theta_1 \leq \bar{\epsilon}_1$ and $\theta_2 \leq \bar{\epsilon}_2$ was never satisfied and the semi-smooth method was not used.

(iii) For two problems, the semi-smooth method was called more than once. In one of these cases, the semi-smooth method always terminated unsuccessfully, while attaining the maximum number of iterations.

(iv) The $\epsilon-$distinct eigenvalues computed by the enumerative and hybrid methods are, in general, the same, but in two examples, the algorithms found two different eigenvalues.

It is important to restate that line-search techniques may be employed within the semi-smooth algorithm that force the algorithm to converge to a stationary point of an appropriate merit function [4, 10]. Stationary points of these functions are only guaranteed to be solutions for the EiCP under monotonicity or similar conditions [4]. We have not investigated incorporating such line-search techniques because the EiCP is a non-monotone complementarity problem, where, in any case, the simple enumerative algorithm acts as a safeguard for the hybrid method.

Finally, Table 5 presents numerical results of experiments that compare the performance of the enumerative method and the hybrid algorithm for solving the QEiCP test problems discussed in [5]. The conclusions of these experiments are similar to the ones presented for the EiCP.

We have also tested using the hybrid method within the parametric algorithm for finding all the eigenvalues. Our experiments revealed that the semi-smooth method tends to converge to the same previously identified complementary eigenvalue. If this eigenvalue does not belong to a particular interval, the hybrid algorithm is attracted toward one of the end-points of this interval that is not an eigenvalue. Consequently, we recommend using the hybrid algorithm only for finding the initial complementary eigenvalue, and then adopting the proposed enumerative method for further iterations.

Table 5: Performance of the enumerative algorithm vs. the hybrid algorithm for solving QEiCPs.

		Enumerative algorithm			Hybrid Algorithm $\theta_1, \theta_2 \leq 10^{-1}$				
Problem	n	Eigenvalue	ITpivot	Nodes	Eigenvalue	ITpivot	Nodes	NTime	Iterations
SeegerAdlyQ(3)	3	0.266	10	0	0.266	10	0	0	0
RAND(0,1,5)	5	0.708	32	0	0.708	32	0	0	0
RAND(0,1,10)	10	−5.575	261	4	0.811	76	0	1	37
RAND(0,1,20)	20	1.114	229	0	1.114	229	0	0	0
RAND(0,1,30)	30	1.055	959	3	1.055	581	0	1	9
RAND(0,1,40)	40	−21.182	2078	4	1.055	935	0	1	30
RAND(0,1,50)	50	1.127	3997	8	−26.020	1001	0	1	26
RAND(0,10,5)	5	−21.760	138	4	−21.760	138	4	0	0
RAND(0,10,10)	10	0.968	211	3	0.968	211	3	0	0
RAND(0,10,20)	20	1.193	484	2	1.193	484	2	1	100^*
RAND(0,10,30)	30	−149.446	2900	8	−149.446	933	2	2	$100^* + 40$
RAND(0,10,40)	40	−197.925	3136	6	−197.925	3136	6	4	$4 \times 100^*$
RAND(0,10,50)	50	−252.888	7605	10	−252.888	1816	2	2	$100^* + 78$
RAND(0,100,5)	5	1.112	46	0	1.112	46	0	0	0
RAND(0,100,10)	10	0.887	99	0	0.887	99	0	0	0
RAND(0,100,20)	20	1.953	5620	38	1.953	5620	38	3	$3 \times 100^*$
RAND(0,100,30)	30	−1494.046	2164	4	−1494.046	1359	3	3	$2 \times 100^* + 53$
RAND(0,100,40)	40	1.077	49894	62	−2032.782	34582	44	23	$22 \times 100^* + 58$
RAND(0,100,50)	50	−2488.298	5186	4	−2488.298	2258	2	2	$100^* + 71$

References

[1] S. ADLY AND A. SEEGER, *A nonsmooth algorithm for cone-constrained eigenvalue problems*, Computational Optimization and Applications, 49 (2011), pp. 299–318.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, 1994.

[3] C. BRÁS, M. FUKUSHIMA, J. JÚDICE, AND S. ROSA, *Variational inequality formulation of the asymmetric eigenvalue complementarity problem and its solution by means of gap functions*, Pacific Journal of Optimization, 8 (2012), pp. 197–215.

[4] F. FACCHINEI AND J. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, 2003.

[5] L. M. FERNANDES, J. J. JÚDICE, H. D. SHERALI, AND M. A. FORJAZ, *On an enumerative algorithm for solving eigenvalue complementarity problems*.

[6] J. JÚDICE, H. SHERALI, AND I. RIBEIRO, *The eigenvalue complementarity problem*, Computational Optimization and Applications, 37 (2007), pp. 139–156.

[7] J. JÚDICE, H. SHERALI, I. RIBEIRO, AND S. ROSA, *On the asymmetric eigenvalue complementarity problem*, Optimization Methods and Software, 24 (2009), pp. 549–586.

[8] H. LE THI, M. MOEINI, T. PHAM DINH, AND J. JÚDICE, *A DC programming approach for solving the symmetric eigenvalue complementarity problem*, Computational Optimization and Applications, 51 (2012), pp. 1097–1117.

[9] Y. S. NIU, T. PHAM, H. A. LE THI, AND J. JÚDICE, *Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem*, to appear in Optimization Methods and Software.

[10] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization, 2nd edition*, Springer, New York, 2006.

[11] A. PINTO DA COSTA, J. MARTINS, I. FIGUEIREDO, AND J. JÚDICE, *The directional instability problem in systems with frictional contacts*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 357–384.

[12] A. PINTO DA COSTA AND A. SEEGER, *Cone-constrained eigenvalue problems: theory and algorithms*, Computational Optimization and Applications, 45 (2010), pp. 25–57.

[13] M. QUEIROZ, J. JÚDICE, AND C. HUMES, *The symmetric eigenvalue complementarity problem*, Mathematics of Computation, 73 (2003), pp. 1849–1863.

[14] N. SAHINIDIS AND M. TAWARMALANI, *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs*, User's Manual, 2005.

[15] A. SEEGER, *Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions*, Linear Algebra and its Applications, 292 (1999), pp. 1–14.

[16] ——, *Quadratic eigenvalue problems under conic constraints*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 700–721.

[17] A. SEEGER AND M. TORKI, *On eigenvalues induced by a cone constraint*, Linear Algebra and its Applications, 372 (2003), pp. 181 – 206.

[18] ——, *Local minima of quadratic forms on convex cones*, Journal of Global Optimization, 44 (2009), pp. 1–28.

[19] A. SEEGER AND J. VICENTE-PÉREZ, *On cardinality of Pareto spectra*, The Electronic Journal of Linear Algebra, 22 (2011), pp. 758–766.

[20] Y. ZHOU AND M. S. GOWDA, *On the finiteness of the cone spectrum of certain linear transformations on Euclidean Jordan algebras*, Linear Algebra and its Applications, 431 (2009), pp. 772–782.