Journal of Optimization Theory and Applications manuscript No.
(will be inserted by the editor)

On the Solution of the Inverse Eigenvalue Complementarity Problem

Carmo P. Brás · Joaquim J. Júdice · Hanif D. Sherali

Received: date / Accepted: date

Abstract In this paper, we discuss the solution of an Inverse Eigenvalue Complementarity Problem. Two nonlinear formulations are presented for this problem. A necessary and sufficient condition for a stationary point of the first of these formulations to be a solution of the problem is established. On the other hand, for assuring global convergence to a solution of this problem when it exists, an enumerative algorithm is designed by exploiting the structure of the second formulation. The use of additional implied constraints for enhancing the efficiency of the algorithm is also discussed. Computational results are provided to highlight the performance of the algorithm.

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

Communicated by Suliman Saleh Al-Homidan.

C. P. Brás (✉)
CMA, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, Caparica, Portugal.
E-mail: mb@fct.unl.pt

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

H. D. Sherali
Grado Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA 24061, USA.
E-mail: hanifs@vt.edu

Mathematics Subject Classification (2000) 65F18 · 90C26 · 90C30

1 Introduction

The Eigenvalue Problem (EP) is currently regarded as one of the three most important problems in Numerical Linear Algebra and finds many applications in several areas of science, engineering, and economics [1–3]. Given a square matrix with real or complex entries, the EP consists of finding a scalar and a vector satisfying a certain equation stated in [1,3]. The scalar is called an eigenvalue and can be a real or a complex number, while the vector is the eigenvector associated with the eigenvalue and may have real or complex components. The number of eigenvalues of the EP is exactly the order of the matrix and many algorithms have been developed to find one, some, or all the eigenvalues and their associated eigenvectors [1–3].

A problem associated with the EP is the so-called Inverse Eigenvalue Problem (IEP) and essentially consists of finding a matrix that conforms to a prescribed set of eigenvalues, and in some cases, eigenvectors. The IEP may take several forms depending on the underlying application, as discussed in detail in the monograph [4]. A number of techniques for the solution of the resulting IEPs are also described in [4], along with the relevance of imposing certain additional specific constraints in defining these problems.

The Eigenvalue Complementarity Problem (ECP) is a generalization of the EP that has been introduced more recently [5] and also arises in many applications [6,7]. Given a square real matrix, the ECP consists of finding a scalar and real non-negative vectors of dimension equal to the matrix order satisfying equality, inequality and complementarity constraints. The components of the vectors are called complementary, hence the name of the problem [8]. A number of algorithms have been introduced for finding a complementary eigenvalue and an associated eigenvector [9–15,7] and, more recently, for computing all the complementary eigenvalues [16]. Some generalizations of this problem have also been discussed in [17–20]. In contrast with the EP, the ECP does not have a known fixed number of eigenvalues. Instead, two upper-bounds for the number of eigenvalues have been derived in [7,21] and in [22] for the symmetric and asymmetric cases, respectively.

An Inverse Eigenvalue Complementarity Problem (IECP) has been recently introduced in [23] and subsequently studied in [24], where some applications of the problem are highlighted. Given a set of complementary eigenvalues, the problem consists of finding a matrix and real non-negative vectors that satisfy the ECP conditions. The IECP may also contain further constraints and may be generalized to other convex cones [24]. Two Newton type approaches for solving the IECP have been proposed in [24]. However, these methods only possess local convergence properties and may fail to find a solution for the IECP in general. Line-search techniques may also be used with these algorithms, but can only guarantee stationary points of appropriate merit functions that may not be solutions to the IECP [25–27].

In this paper, we address the IECP as a global optimization problem. A first proposed formulation NLP_1 of the IECP is introduced such that IECP has a solution if and only if NLP_1 has a global minimum with a zero objective function value. If the eigenvectors are also given, then NLP_1 is a convex quadratic program that can be solved in polynomial time [27]. However, in this paper, we assume that the eigenvectors have to be computed. A necessary and sufficient condition for a stationary point of NLP_1 to be a solution to the IECP is established in the sequel, but the IECP requires a global optimization algorithm to be solved in general. Accordingly, to assure finding a solution to IECP when it exists, we consider a second formulation NLP_2 that modifies NLP_1 by introducing additional variables.

As before, IECP has a solution if and only if NLP_2 has a global minimum with a zero objective function value. An enumerative method that assures global convergence is proposed in this paper in order to find such a global minimum. The algorithm works with bounding intervals for each component of the eigenvectors. For each node of the enumerative tree, given an associated set of such intervals, a stationary point of NLP_2 is computed, for which, either the objective function value is zero and a solution to the IECP is therefore at hand, or else, two new nodes are generated by partitioning a chosen interval into two new ones. Lower and upper bounds for the entries of the matrix and some further constraints based on the Reformulation-Linearization Technique (RLT) [28] are added to NLP_2 for guaranteeing global convergence of the algorithm to a solution to IECP, when it exists. Furthermore, the algorithm employs some judicious rules for selecting the node and the interval for partitioning at each iteration in order to enhance its computational efficiency, besides facilitating global convergence.

A number of elementary techniques for computing lower and upper bounds for the entries of the matrix and for computing a stationary point of NLP_2 from a stationary point of NLP_1 are also introduced to enhance the computational efficiency of the algorithm. In some instances, it is important to add some further constraints to NLP_2 so that the matrix conforms to a desired structure, which also facilitates the solution of the underlying IECP. Such restrictions include requirements for the matrix to be non-negative or symmetric, or for the complementary eigenvalues to be ordinary eigenvalues of principal submatrices.

Computational results are presented to demonstrate that the algorithm is quite effective in solving IECPs of the form generated, according to the process discussed in [24]. In fact, the algorithm is able to solve all these IECPs relatively easily. In a second experiment, we considered ECPs with special symmetric matrices as discussed in [7]. For this matrix, the number of complementary eigenvalues of the ECP is known, each one being an ordinary eigenvalue of a principal submatrix of the original matrix. We tried to solve IECPs of order less than or equal to five for such complementary eigenvalues with and without further constraints that enforce the non-negativity and symmetry for the matrix and require the complementary eigenvalues to satisfy the property stated above. The algorithm was only able to solve the case of order two when just the non-negativity and symmetry conditions were required, but could solve very efficiently IECPs of order less than or equal to five when the remaining constraints involving the principal submatrices of the original matrix were also added to the nonlinear programs NLP_1 and NLP_2.

The remainder of this paper is organized as follows. In Section 2, we analyze the formulation NLP_1. The formulation NLP_2 and the enumerative algorithm are discussed in Section 3. Some advanced techniques for enhancing the computational efficiency of the algorithm are described in Section 4. Computational experimental results are reported in Section 5, and the paper concludes with some closing remarks in Section 6.

2 A Nonlinear Programming Formulation

Given a square matrix $A \in \mathbb{R}^{n\times n}$, the ECP consists of finding a scalar $\lambda \in \mathbb{R}$ and a vector $x \in \mathbb{R}^n$ such that

$$\text{ECP:} \qquad \{w = Ax - \lambda x,\ x \geqslant 0,\ w \geqslant 0,\ \langle x, w\rangle = 0,\ \langle e, x\rangle = 1\},$$

where $w \in \mathbb{R}^n$, and $e \in \mathbb{R}^n$ is a vector of ones. Note that the constraint $\langle e, x\rangle = 1$ has been added without any loss of generality to avoid the null vector to be a solution of the ECP. Due to the non-negativity of the variables x_i and w_i, the complementarity condition $\langle x, w\rangle = 0$ means that $x_i = 0$ or $w_i = 0$ for each $i \in \mathbb{N}_n$.

As the name suggests, the IECP consists of finding a matrix A and vectors $x^k \in \mathbb{R}^n$ and $w^k \in \mathbb{R}^n$, $k \in \mathbb{N}_p$, for a given set of p complementary eigenvalues λ_k, $k \in \mathbb{N}_p$, such that

$$\text{IECP:} \qquad \{w^k = Ax^k - \lambda_k x^k,\ x^k \geqslant 0,\ w^k \geqslant 0,\ \langle x^k, w^k\rangle = 0,\ \langle e, x^k\rangle = 1\},$$

for $k \in \mathbb{N}_p$. Since this problem is trivial for $p \leqslant n$ (reducing to the IEP with $w^k \equiv 0,\ \forall\, k \in \mathbb{N}_p$), we assume that $p > n$ in this paper.

A first nonlinear formulation to the IECP is given as follows:

$$\text{NLP}_1: \qquad \text{Minimize } f_1(x, w, A) := \sum_{i=1}^{n}\sum_{k=1}^{p}\left((\psi_i^k)^2 + x_i^k w_i^k\right)$$
$$\text{subject to } \{w_i^k \geqslant 0,\ x_i^k \geqslant 0,\ \langle e, x^k\rangle = 1,\ i \in \mathbb{N}_n,\ k \in \mathbb{N}_p\},$$

where

$$\psi_i^k := w_i^k + \lambda_k x_i^k - \sum_{j=1}^{n} a_{ij} x_j^k,\ i \in \mathbb{N}_n,\ k \in \mathbb{N}_p \tag{1}$$

and a_{ij} are also decision variables.

We now investigate when a stationary point of NLP_1 is a global minimum of NLP_1, that is, a solution to IECP (note that ψ_i^k in (1) is used only as an abbreviated notation and is not actually a new auxiliary decision variable added to the problem).

Theorem 2.1

(i) For any stationary point of NLP$_1$, we have

$$\sum_{i=1}^{n}\left((\psi_i^k)^2 + x_i^k w_i^k\right) = \frac{\gamma_k}{2}, \;\; k \in \mathbb{N}_p,$$

where ψ_i^k are given by (1) and γ_k are the Lagrange multipliers associated with the constraints $\langle e, x^k\rangle = 1, \quad k \in \mathbb{N}_p$.

(ii) A stationary point of NLP$_1$ is a solution to IECP if and only if $\gamma_k = 0$ for all $k \in \mathbb{N}_p$.

Proof: A stationary point of NLP$_1$ satisfies:

$$2\,\psi_i^k + x_i^k = \alpha_i^k, \qquad i \in \mathbb{N}_n;\; k \in \mathbb{N}_p, \tag{2}$$

$$2\,\psi_i^k(\lambda_k - a_{ii}) - 2\sum_{l\neq i}\psi_l^k a_{li} + w_i^k = \gamma_k + \beta_i^k, \qquad i \in \mathbb{N}_n;\; k \in \mathbb{N}_p, \tag{3}$$

$$\sum_{k=1}^{p}\psi_i^k x_j^k = 0, \qquad i \in \mathbb{N}_n,\, j \in \mathbb{N}_n,$$

$$\alpha_i^k \geqslant 0,\, w_i^k \geqslant 0,\, \alpha_i^k w_i^k = 0, \qquad i \in \mathbb{N}_n;\; k \in \mathbb{N}_p, \tag{4}$$

$$\beta_i^k \geqslant 0,\, x_i^k \geqslant 0,\, \beta_i^k x_i^k = 0, \qquad i \in \mathbb{N}_p;\; k \in \mathbb{N}_p, \tag{5}$$

$$\sum_{j=1}^{n} x_j^k = 1 \qquad k \in \mathbb{N}_p, \tag{6}$$

where α_i^k, β_i^k, and γ_k are Lagrange multipliers associated with the constraints $w_i^k \geqslant 0$, $x_i^k \geqslant 0$, and $\langle e, x^k\rangle = 1$, respectively.

For all $i \in \mathbb{N}_n,\, k \in \mathbb{N}_p$, (3) above is given by $2\,\lambda_k\psi_i^k - 2\sum_{l=1}^{n} a_{li}\psi_l^k + w_i^k = \gamma_k + \beta_i^k$. Denoting the vectors $\psi^k := (\psi_i^k) \in \mathbb{R}^n$, $\beta^k := (\beta_i^k) \in \mathbb{R}^n$, and $\alpha^k := (\alpha_i^k) \in \mathbb{R}^n$, we can rewrite this as follows:

$$2\,\lambda_k\psi^k - 2A^T\psi^k + w^k = \gamma_k e + \beta^k. \tag{7}$$

On the other hand, by (2) and (4), we have $2\,\psi_i^k w_i^k + x_i^k w_i^k = 0,\; i \in \mathbb{N}_n,\; k \in \mathbb{N}_p$, i.e.,

$$\langle x^k, w^k\rangle + 2\langle\psi^k, w^k\rangle = 0. \tag{8}$$

Multiplying (7) by $(x^k)^T$, we have upon using (5) and (6) that

$$2\,\lambda_k\langle x^k,\psi^k\rangle - 2\langle Ax^k,\psi^k\rangle + \langle x^k,w^k\rangle = \gamma_k,$$

which upon adding (8) and noting (1) yields $\left\|\psi^k\right\|^2 + \langle x^k,w^k\rangle = \frac{\gamma_k}{2}$. This establishes Part (i). Part (ii) now follows by noting that a feasible solution to Problem NLP_1 solves IECP if and only if the objective function value is zero. □

This theorem shows how a stationary point of NLP_1 can possibly yield a solution to the IECP, and provides insights into what makes this occur in practice. Indeed, this occurs quite frequently as borne by our computational experiments reported in Section 5. Yet, a global optimization algorithm is required to deal with the IECP in general. This is discussed in the next section.

3 An Enumerative Method

Consider the nonlinear programming formulation NLP_1 of the IECP and introduce pn^2 additional variables y_{ij}^k that represent the product relationship:

$$y_{ij}^k := a_{ij}x_j^k \tag{9}$$

for all $i,j\in\mathbb{N}_n$, and $k\in\mathbb{N}_p$. This leads to the following alternative nonlinear programming formulation for the IECP:

$$\mathrm{NLP}_2: \qquad \text{Minimize} \sum_{k=1}^{p}\sum_{i=1}^{n}\left[\sum_{j=1}^{n}(y_{ij}^k - a_{ij}x_j^k)^2 + x_i^k w_i^k\right] \tag{10}$$

$$\text{subject to } \sum_{j=1}^{n} y_{ij}^k - \lambda_k x_i^k = w_i^k,\ i\in\mathbb{N}_n,\ k\in\mathbb{N}_p, \tag{11}$$

$$\sum_{j=1}^{n} x_j^k = 1, \qquad k\in\mathbb{N}_p, \tag{12}$$

$$x_j^k \geqslant 0,\ w_j^k \geqslant 0, \qquad j\in\mathbb{N}_n,\ k\in\mathbb{N}_p, \tag{13}$$

where y_{ij}^k and a_{ij} are also decision variables. The basic idea of the enumerative method proposed in the sequel is to achieve convergence to a solution to the inverse eigenvalue problem by iteratively partitioning the intervals of the x_j^k-variables. Accordingly, assume that we have the following general bounds on the x_j^k-variables at any stage of this enumerative process:

$$0 \leqslant l_j^k \leqslant x_j^k \leqslant u_j^k \leqslant 1,\ j \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{14}$$

where, to begin with, we have $l_j^k = 0$ and $u_j^k = 1,\ \forall j, k$. Furthermore, let θ_1 and θ_2 be real numbers such that $\theta_1 \leqslant a_{ij} \leqslant \theta_2$ for all i, j. A procedure for estimating $\theta_i, i = 1, 2$, is discussed in Section 4. Given the bounds (14) on the x_j^k-variables along with the assumed restrictions on the a_{ij}-variables, we construct the following RLT bound-factor constraints [28] in order to help induce (9):

$$\begin{aligned}
&\text{(a)}\quad l_j^k(a_{ij} - \theta_1) \leqslant y_{ij}^k - \theta_1 x_j^k \leqslant u_j^k(a_{ij} - \theta_1),\ \forall i, j, k\\
&\text{(b)}\quad l_j^k(\theta_2 - a_{ij}) \leqslant \theta_2 x_j^k - y_{ij}^k \leqslant u_j^k(\theta_2 - a_{ij}),\ \forall i, j, k.
\end{aligned}$$

With this construct, the nonlinear formulation NLP$_2$ for the IECP can be rewritten as follows:

$$\text{NLP}_2 : \text{Minimize } f_2(x, y, w, A) := \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^p (y_{ij}^k - a_{ij} x_j^k)^2 + \sum_{i=1}^n \sum_{k=1}^p x_i^k w_i^k \tag{15}$$

subject to

$$\sum_{j=1}^n y_{ij}^k - \lambda_k x_i^k = w_i^k, \qquad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{16}$$

$$\sum_{j=1}^n x_j^k = 1, \qquad k \in \mathbb{N}_p, \tag{17}$$

$$l_j^k(a_{ij} - \theta_1) \leqslant y_{ij}^k - \theta_1 x_j^k \leqslant u_j^k(a_{ij} - \theta_1), \qquad i,\ j \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{18}$$

$$l_j^k(\theta_2 - a_{ij}) \leqslant \theta_2 x_j^k - y_{ij}^k \leqslant u_j^k(\theta_2 - a_{ij}), \qquad i,\ j \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{19}$$

$$\theta_1 \leqslant a_{ij} \leqslant \theta_2, \qquad i,\ j \in \mathbb{N}_n, \tag{20}$$

$$x_j^k, w_j^k \geqslant 0, \qquad j \in \mathbb{N}_n,\ k \in \mathbb{N}_p. \tag{21}$$

Observe that (x, y, w, A) provides a solution to the IECP if and only if it solves Problem NLP$_2$ with a zero objective value. Accordingly, the proposed enumerative method seeks a global minimum of NLP_2. To do this, a binary tree is generated based on partitioning the intervals $[l_j^k, u_j^k]$, where l_j^k and u_j^k satisfy (14). Each node of the enumeration tree has a set of such intervals associated with it, one for each x-variable. At each node, a stationary point $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ for NLP_2 is computed. If the objective function value for this solution is zero, then a solution to the IECP is at hand (i.e., we have obtained a global minimum to NLP_2). Otherwise, two new nodes are generated by branching on a variable (i.e, by partitioning its corresponding interval), which is selected as one that yields the maximum discrepancy index ξ_i^k at the stationary point $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$, where

$$\xi_i^k := \sum_{j=1}^{n} |\bar{y}_{ji}^k - \bar{a}_{ji}\bar{x}_i^k|, \ \forall i, k.$$

A special heuristic rule for selecting a node at each iteration and some logical tests (which are (partly) essential for the convergence proof) complete the steps of the enumerative method. The detailed formal steps of this algorithm are presented below.

Enumerative Algorithm

Step 0. Let ϵ, ϵ_1, and ϵ_2 be positive tolerances. Set $t = 1, l_j^k = 0, u_j^k = 1, \forall\,(j,k)$, and find a stationary point $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{a})$ for NLP_2. If $f_2(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{A}) \leqslant \epsilon$, then stop with $A = [\tilde{a}_{ij}]$ as a solution to IECP (within the tolerance ϵ). Otherwise, let $\mathcal{L} = \{1\}$ be the set of open nodes, set $UB(1) := f_2(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{A})$, and let $N = 1$ be the number of nodes generated.

Step 1. If $\mathcal{L} = \emptyset$, terminate; the given IECP problem has no solution. Otherwise, select a node $t \in \mathcal{L}$ such that

$$UB(t) := \min\{UB(i) : i \in \mathcal{L}\},$$

and let $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{A})$ be the stationary point for Problem NLP_2 that was found at this node.

Step 2. Let

$$\xi_i^{1k} := \sum_{j=1}^{n} |\tilde{y}_{ji}^k - \tilde{a}_{ji}\tilde{x}_i^k| \quad \text{and} \quad \xi_i^{2k} := \tilde{x}_i^k \tilde{w}_i^k, \ \forall i, k. \tag{22}$$

Let $Q_1 := \max_{(i,k)}\{\xi_i^{1k}\}$ and $Q_2 := \max_{(i,k)}\{\xi_i^{2k}\}$. If $Q_1 \leqslant \epsilon_1$ and $Q_2 \leqslant \epsilon_2$, then stop with $A = [\tilde{a}_{ij}]$ as a solution to IECP (within the tolerances (ϵ_1, ϵ_2)). Otherwise, let

$$(i^*, k^*) \in \arg\max_{(i,k)} \left\{ \frac{\xi_i^{1k}}{\max\{Q_1, \epsilon_1\}} + \frac{\xi_i^{2k}}{\max\{Q_2, \epsilon_2\}} \right\}, \tag{23}$$

and partition the current interval $[l_{i^*}^{k^*}, u_{i^*}^{k^*}]$ of $x_{i^*}^{k^*}$ into $[l_{i^*}^{k^*}, \hat{x}_{i^*}^{k^*}]$ and $[\hat{x}_{i^*}^{k^*}, u_{i^*}^{k^*}]$ to generate two nodes from node t, namely, $N+1$ and $N+2$, where

$$\hat{x}_{i^*}^{k^*} := \begin{cases} \tilde{x}_{i^*}^{k^*}, & \text{if } \min\{\tilde{x}_{i^*}^{k^*} - l_{i^*}^{k^*}, u_{i^*}^{k^*} - \tilde{x}_{i^*}^{k^*}\} \geqslant 0.1(u_{i^*}^{k^*} - l_{i^*}^{k^*}) \\ \dfrac{l_{i^*}^{k^*} + u_{i^*}^{k^*}}{2}, & \text{otherwise.} \end{cases} \tag{24}$$

Step 3. For each node $s = N+1$ and $s = N+2$, let $[l_j^k, u_j^k]$ denote the bounding interval for x_j^k at the node, and do the following:

1. If $\sum_{j=1}^{n} u_j^k < 1$, or $\sum_{j=1}^{n} l_j^k > 1$, for any k : then this subproblem is infeasible; if $s = N+1$, then proceed to the next node in the loop of Step 3 (or directly to Step 4 if both nodes have been processed).

2. (i) For any k, j : If $l_j^k > 0 \Rightarrow$ set $w_j^k = 0$.
   (ii) For any k : If $\sum_{j=1}^{n} u_j^k = 1$, then fix $x_j^k = u_j^k$, $\forall j$, and also let $l_j^k \equiv u_j^k, \forall j$.
   (iii) For any k : If $\sum_{j=1}^{n} l_j^k = 1$, then fix $x_j^k = l_j^k$, $\forall j$, and also let $u_j^k \equiv l_j^k, \forall j$.

3. Find a stationary point $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{A})$ of NLP$_2$ for the subproblem at node s. If $f_2(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{A}) \leqslant \epsilon$ then stop with a solution $A = [\tilde{a}_{ij}]$ for IECP.

Step 4. $\mathcal{L} = \mathcal{L} \setminus \{t\}$ and return to Step 1.

The following theorem establishes the global convergence of the proposed enumerative algorithm:

Theorem 3.1 *The enumerative algorithm for NLP$_2$ run with $\epsilon_1 = \epsilon_2 = 0$ either terminates finitely with a solution to IECP (possibly indicating that no solution exists), or else, an infinite branch-and-bound (B&B) tree is generated such that along any infinite branch of this tree, any accumulation point of the stationary points obtained for NLP$_2$ solves IECP.*

Proof: The case of finite termination is obvious (including the detection that no solution exists). Hence, suppose that an infinite B&B tree is generated, and consider any infinite branch. Denote the vector $\zeta = (x, y, w, a)$, and let any accumulation point of the stationary points obtained for NLP$_2$ along this branch, as corresponding to a sequence of indices $s \in S$, yield $\{\zeta^s\} \to \zeta^*$ and $\{[l^{ks}, u^{ks}]\}_s \to [l^{k*}, u^{k*}],\ \forall\, k \in \mathbb{N}_p$, where $[l^{ks}, u^{ks}]$ denotes the vector of bounds on x^k at node s of the B&B tree, $\forall\, k \in \mathbb{N}_p,\ s \in S$. We will show that ζ^* yields a solution to IECP.

Note that, along the infinite branch under consideration, there exists some index-pair $(\hat{i}, \hat{k})$ such that we branch on the interval for $x_{\hat{i}}^{\hat{k}}$ infinitely often. Let this correspond to nodes indexed by $s \in S_1 \subseteq S$. By the partitioning rule (24), since the interval length for $x_{\hat{i}}^{\hat{k}}$ decreases by a geometric ratio of at most 0.9 over $s \in S_1$, we have in the limit that

$$l_{\hat{i}}^{\hat{k}*} = u_{\hat{i}}^{\hat{k}*} = x_{\hat{i}}^{\hat{k}*} = \nu^*, \text{ say.} \tag{25}$$

Observe that we also have in the limit that

$$x_{\hat{i}}^{\hat{k}*} . w_{\hat{i}}^{\hat{k}*} = 0, \tag{26}$$

since either $x_{\hat{i}}^{\hat{k}*} = 0$, or else, if $x_{\hat{i}}^{\hat{k}*} > 0$, then by the logical tests at Step 3, for $s \in S_1$ large enough, we have by (25) that $w_{\hat{i}}^{\hat{k}s} \equiv 0$, whence $w_{\hat{i}}^{\hat{k}*} = 0$.

Furthermore, from (25) and the RLT bound-factor constraints (18) and (19), we have in the limit that

$$\nu^*(a_{j\hat{i}}^* - \theta_1) \leqslant y_{j\hat{i}}^{\hat{k}*} - \theta_1 \nu^* \leqslant \nu^*(a_{j\hat{i}}^* - \theta_1),\ \forall\, j \in \mathbb{N}_n \tag{27}$$

and

$$\nu^*(\theta_2 - a_{j\hat{i}}^*) \leqslant \theta_2 \nu^* - y_{j\hat{i}}^{\hat{k}*} \leqslant \nu^*(\theta_2 - a_{j\hat{i}}^*),\ \forall\, j \in \mathbb{N}_n, \tag{28}$$

where we have interchanged the indices i and j to suit (25). The constraints (27) and (28) reduce to $\nu^* a_{j\hat{i}}^* \leqslant y_{j\hat{i}}^{\hat{k}*} \leqslant \nu^* a_{j\hat{i}}^*,\ \forall\, j \in \mathbb{N}_n$, i.e., by (25),

$$y_{j\hat{i}}^{\hat{k}*} = a_{j\hat{i}}^* x_{\hat{i}}^{\hat{k}*},\ \forall\, j \in \mathbb{N}_n. \tag{29}$$

Thus, from (26) and (29), we have that in the limit in (22), as $s \to \infty$ with $s \in S_1$, the entities

$$\xi_{\hat{i}}^{1\hat{k}*} = \xi_{\hat{i}}^{2\hat{k}*} = 0.$$

However, by the selection of the index-pair $(\hat{i}, \hat{k})$ for $s \in S$, via (23), we get in the limit as $s \to \infty$, $s \in S_1$ that $\xi_i^{1k} = \xi_i^{2k} = 0$, $\forall\, i, k$, i.e.,

$$y_{ji}^{k*} = a_{ji}^* x_i^{k*}, \quad \text{and} \quad x_i^{k*}.w_i^{k*} = 0,\, \forall\, i, j \in \mathbb{N}_n,\, k \in \mathbb{N}_p. \tag{30}$$

Consequently, the set of constraints (16) yield from (30) that

$$\sum_{j=1}^{n} a_{ij}^* x_j^{k*} - \lambda_k x_i^{k*} = w_i^{k*},\, \forall\, i \in \mathbb{N}_n,\, k \in \mathbb{N}_p, \tag{31}$$

and the remaining constraints (17)–(21) yield

$$\theta_1 \leqslant a_{ij}^* \leqslant \theta_2, \forall\, i, j, \text{ and } x^{k*} \geqslant 0, w^{k*} \geqslant 0, \text{ and } \langle e, x^{k*} \rangle = 1, \forall\, k \in \mathbb{N}_p. \tag{32}$$

Thus, (30)–(32) imply that ζ^* represents a solution to IECP. □

Remark 3.1 Note that, by the proof of Theorem 3.1, only one set of the two RLT bound-factor constraints (18)–(19) are necessary to assure convergence to a solution to IECP. However, we retain the pair of constraints since they better guide the algorithm to converge more efficiently.

4 Improving Computational Efficiency

4.1 Computing Stationary Points for NLP$_2$

In the previous section, we have shown that a stationary point $(\bar{x}, \bar{w}, \bar{A})$ of NLP$_1$ may provide a solution to the IECP. As reported in Section 5, our computational experiments showed that, in practice, such a stationary point may in many cases solve the IECP. Accordingly, we recommend computing such a stationary point at the root node of the enumerative algorithm. If such a stationary point does not yield a solution to the IECP, then it would be computationally beneficial if a stationary point for NLP$_2$ could be computed from the available stationary point for NLP$_1$ without much additional effort. In this section, we investigate whether this is possible, firstly at the root node, and later at any node of the tree.

Consider Problem NLP$_2$ given by (10)-(13), without the additional constraints (18)-(20), and for the sake of clarity, call this relaxed (yet valid) formulation $\overline{NLP}_2$. Furthermore, suppose that $(\bar{x}, \bar{w}, \bar{A})$ is a stationary point for NLP$_1$. Now, consider the following np programs:

$$\text{Minimize} \left\{ \sum_{j=1}^{n} (y_{ij}^k - \bar{a}_{ij}\bar{x}_j^k)^2 : \sum_{j=1}^{n} y_{ij}^k = \bar{w}_i^k + \lambda_k \bar{x}_i^k \right\}, \tag{33}$$

for $i \in \mathbb{N}_n, \; k \in \mathbb{N}_p$. Each one of these programs can be written in the following abbreviated form:

$$\text{Minimize} \left\{ c_0 - 2\langle c, y\rangle + \langle y, y\rangle : \langle e, y\rangle = b_0 \right\}.$$

This optimization problem is a strictly convex quadratic program that has a unique optimal solution $\bar{y}$ satisfying $\{2\bar{y} - 2c = \lambda e, \; \langle e, \bar{y}\rangle = b_0\}$, where λ is the multiplier associated with the linear constraint. Thus, the unique optimal solution is given by

$$\bar{y} = c + \frac{b_0 - \langle e, c\rangle}{\langle e, e\rangle}\, e.$$

Applying this result to the np programs (33), we have that the corresponding optimal solution is given by

$$\bar{y}_{ij}^k := \bar{a}_{ij}\bar{x}_j^k + \frac{1}{n}(\bar{w}_i^k + \lambda_k \bar{x}_i^k - \sum_{l=1}^{n} \bar{a}_{il}\bar{x}_l^k). \tag{34}$$

The next theorem shows that $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ is a stationary point for the nonlinear program $\overline{NLP}_2$.

Theorem 4.1 *If* $(\bar{x}, \bar{w}, \bar{A})$ *is a stationary point for NLP*$_1$ *and* $\bar{y}$ *is given by* (34)*, then* $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ *is a stationary point for* $\overline{NLP}_2$.

Proof: Consider the program NLP$_1$. Let $\alpha_i^k, \beta_i^k, \eta_i^k, \gamma_k$ be the Lagrange multipliers associated with the constraints $w_i^k \geqslant 0$, $x_i^k \geqslant 0$, $\sum_{j=1}^{n} y_{ij}^k - \lambda_k x_i^k - w_i^k = 0$, and $\sum_{j=1}^{n} x_j^k = 1$, respectively. If

$$\rho_{ij}^k := y_{ij}^k - a_{ij}x_j^k,$$

then any stationary point for this problem satisfies the following conditions:

$$x_i^k = -\eta_i^k + \alpha_i^k, \quad \alpha_i^k w_i^k = 0, \quad \alpha_i^k \geqslant 0, \quad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{35}$$

$$2\sum_{l=1}^n \rho_{li}^k(-a_{li}) + w_i^k = -\lambda_k\eta_i^k + \gamma_k + \beta_i^k,\ \ \beta_i^k x_i^k = 0,\ \ \beta_i^k \geqslant 0, i \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{36}$$

$$2\rho_{ij}^k = \eta_i^k, \quad i,\, j \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{37}$$

$$-2\sum_{k=1}^p \rho_{ij}^k x_j^k = 0, \quad i,\, j \in \mathbb{N}_n. \tag{38}$$

By using (37), we can rewrite (35), (36), and (38) as follows:

$$x_i^k + \eta_i^k = \alpha_i^k, \quad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p,$$

$$\eta_i^k(\lambda_k - a_{ii}) - \sum_{l \neq i} a_{li}\,\eta_i^k + w_i^k = \gamma_k + \beta_i^k, i \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{39}$$

$$\sum_{k=1}^p \eta_i^k x_j^k = 0, \quad i,\, j \in \mathbb{N}_n, \tag{40}$$

$$\alpha_i^k \geqslant 0, \quad \beta_i^k \geqslant 0, \quad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p, \tag{41}$$

$$\alpha_i^k w_i^k = \beta_i^k x_i^k = 0, \quad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p. \tag{42}$$

Now, if $(\bar{x}, \bar{w}, \bar{A})$ is a stationary point for NLP_1 and $\bar{y}$ is given by (34), then $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ satisfies the conditions (39)–(42), with $\alpha_i^k,\ \beta_i^k,\ \gamma_k$ being the Lagrange multipliers associated with the stationary point of NLP_1, and with $2\eta_i^k = \psi_i^k,\ i \in \mathbb{N}_n,\ k \in \mathbb{N}_p$, where ψ_i^k is given by (1). This set of conditions implies that $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ is a stationary point for $\overline{NLP}_2$. □

For this stationary point $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$, we have

$$\begin{aligned} f_2(\bar{x}, \bar{y}, \bar{w}, \bar{A}) &= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^p (\bar{y}_{ij}^k - \bar{a}_{ij}\bar{x}_j^k)^2 + \sum_{i=1}^n \sum_{k=1}^p \bar{x}_i^k \bar{w}_i^k \\ &= \sum_{i=1}^n \sum_{k=1}^p \left[\sum_{j=1}^n (\bar{y}_{ij}^k - \bar{a}_{ij}\bar{x}_j^k)^2 + \bar{x}_i^k \bar{w}_i^k \right] \end{aligned}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{p}\left(\frac{1}{n^2}\sum_{j=1}^{n}\left(\bar{w}_i^k + \lambda_k \bar{x}_i^k - \sum_{j=1}^{n}\bar{a}_{ij}\bar{x}_j^k\right)^2 + \bar{x}_i^k \bar{w}_i^k\right)$$
$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\left(\bar{w}_i^k + \lambda_k \bar{x}_i^k - \sum_{j=1}^{n}\bar{a}_{ij}\bar{x}_j^k\right)^2 + \sum_{i=1}^{n}\sum_{k=1}^{p}\bar{x}_i^k \bar{w}_i^k.$$

Consequently any stationary point $(\bar{x}, \bar{w}, \bar{A})$ for NLP$_1$ gives a stationary point $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ for $\overline{NLP}_2$, with $\bar{y}$ given by (34), and with $f_2(\bar{x}, \bar{y}, \bar{w}, \bar{A}) = 0$ if and only if $f_1(\bar{x}, \bar{w}, \bar{A}) = 0$. Now, suppose that we are at any node t of the tree generated by the enumerative method, and consider the following nonlinear program NLP$_1(t)$ that is obtained from NLP$_1$ by adding the bounds on the entries of A and the intervals for the variables x_i^k associated with the node t:

$$\begin{aligned}
\text{NLP}_1(t):\ \text{Minimize}\ & f_1(x, w, A) := \sum_{i=1}^{n}\sum_{k=1}^{p}[(w_i^k + \lambda_k x_i^k - \sum_{j=1}^{n} a_{ij}x_j^k)^2 + x_i^k w_i^k] \\
\text{subject to}\ & \sum_{j=1}^{n} x_j^k = 1, \quad k \in \mathbb{N}_p \\
& \theta_1 \leqslant a_{ij} \leqslant \theta_2,\ i \in \mathbb{N}_n,\ j \in \mathbb{N}_n \\
& l_i^k \leqslant x_i^k \leqslant u_i^k,\ i \in \mathbb{N}_n,\ k \in \mathbb{N}_p \\
& w_i^k \geqslant 0, \qquad i \in \mathbb{N}_n,\ k \in \mathbb{N}_p.
\end{aligned}$$

Furthermore, given the stationary point $(\bar{x}, \bar{w}, \bar{A})$ for NLP$_1(t)$ at this node, assume that $\theta_1 < \bar{a}_{ij} < \theta_2$, and define the following sets associated with $\bar{x} = [x_i^k] \in \mathbb{R}^{np}$:

$$\begin{aligned}
J^k &:= \{j : l_j^k < \bar{x}_j^k < u_j^k\},\ k \in \mathbb{N}_p \\
L^k &:= \{j : \bar{x}_j^k = l_j^k\},\ k \in \mathbb{N}_p \\
U^k &:= \{j : \bar{x}_j^k = u_j^k\},\ k \in \mathbb{N}_p.
\end{aligned}$$

Let $\bar{y} = [y_{ij}^k] \in \mathbb{R}^{n^2 p}$ be accordingly defined by (for $i \in \mathbb{N}_n$, and $k \in \mathbb{N}_p$)

$$\begin{aligned}
\bar{y}_{ij}^k &:= \bar{a}_{ij} l_j^k,\ j \in L^k \\
\bar{y}_{ij}^k &:= \bar{a}_{ij} u_j^k,\ j \in U^k,
\end{aligned}$$

and let $\bar{y}_{ij}^k$ for $j \in J^k$ be the unique optimal solution of the np programs:

$$\text{Minimize}\left\{\sum_{j \in J^k}(y_{ij}^k - \bar{a}_{ij}\bar{x}_j^k)^2 :\ \sum_{j \in J^k} y_{ij}^k = \bar{w}_i^k + \lambda_k \bar{x}_i^k - \sum_{j \in (L^k \cup U^k)} \bar{y}_{ij}^k\right\}.$$

As per (34), we thus have for $i \in \mathbb{N}_n$, $k \in \mathbb{N}_p$, and $j \in J^k$,

$$\bar{y}_{ij}^k := \bar{a}_{ij}\bar{x}_j^k + \frac{1}{|J^k|}(\bar{w}_i^k + \lambda_k \bar{x}_i^k - \sum_{l=1}^{n} \bar{a}_{il}\bar{x}_l^k),$$

where $|J^k|$ represents the number of elements in the set J^k.

Similarly to the proof of Theorem 4.1, it is easy to show that $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ is a stationary point of NLP$_2$ given by (15)-(21) at node t and $f_2(\bar{x}, \bar{y}, \bar{w}, \bar{A}) = 0$ if and only if $f_1(\bar{x}, \bar{w}, \bar{A}) = 0$. If this is not the case, then it is necessary to verify whether $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ is feasible for NLP$_2$. This usually occurs in our computational experience and $(\bar{x}, \bar{y}, \bar{w}, \bar{A})$ yields a stationary point for NLP$_2$ at node t. If this is not the case, then a new stationary point for NLP$_2$ is computed from scratch.

4.2 Bounds for the Matrix A

The enumerative algorithm requires lower and upper bounds θ_1 and θ_2 for all the variables a_{ij} of the solution matrix A of the IECP. Too large absolute values for these bounds might induce numerical problems for the local solver used to find stationary points for NLP$_1(t)$ and NLP$_2$. In this section, we discuss a good heuristic (but theoretically valid in some special cases as indicated below) choice for these lower and upper bounds. We also show by an example the benefit of this choice in practice.

Taking into consideration the structure of the ECP, we heuristically propose the following choices for θ_1 and θ_2:

$$\theta_2 := \delta \times \max_{k \in \mathbb{N}_p} |\lambda_k|, \quad \text{and} \quad \theta_1 := -\theta_2, \tag{43}$$

(or $\theta_1 = 0$, when A is non-negative), where δ is a positive real number such that $\delta \geq n$.

Based on a result presented in [23], one of the referees suggested that for $n < p \leq \dfrac{n(n+1)}{2}$, the following values of θ_i are valid:

$$\theta_1 := \min\{0, \lambda_1\}, \text{ and } \theta_2 := \max\{\lambda_n, \lambda_p - \lambda_1\}, \tag{44}$$

where the given eigenvalues λ_i are assumed to be in nondecreasing order. We therefore use the bounds (44) whenever the stated condition holds true.

4.3 Additional Constraints

The numerical resolution of an IECP is a challenging task even for low dimensional instances. In this section, we explore the cases where the matrix A has special properties that can be further exploited.

(i) **Non-negative matrix A:**

When we know that there exists a solution to the IECP with A being a non-negative matrix, we can reduce the range bounds for the matrix A and consider $\theta_1 = 0$.

(ii) **Symmetric matrix A:**

Numerical experiments to be reported in Section 5 include cases involving symmetric matrices A. For these instances, the additional set of constraints $a_{ij} = a_{ji}, \forall\, i \neq j$, can be included in the formulation NLP_2 to reduce the complexity of the problem.

(iii) **Fixing diagonal elements:**

Consider a particular problem where the set of eigenvalues $\lambda_k, k \in \mathbb{N}_p$, includes the diagonal elements of the matrix. By fixing the diagonal elements of A, i.e., $a_{ii} = \lambda_i, i \in \mathbb{N}_n$ and imposing the constraints $a_{ij} \geqslant 0$, $\forall i \neq j$, the complexity of the problem can be decreased, since the vectors x^k and w^k corresponding to the values of the diagonal are easily computed by letting x^k equal the corresponding unit vector, and so the effective cardinality of the set of eigenvalues reduces to $p - n$.

(iv) **Complementary eigenvalues being classic eigenvalues of principal submatrices:**

Since the complementary eigenvalues are always ordinary eigenvalues of principal submatrices of A [16], we can include valid constraints on x_i^k and w_i^k in order to reduce the complexity of the problem when we know the principal submatrices of A that correspond to these eigenvalues. This is achieved by introducing the constraints

$$\left\{ \sum_{j \in I_k} x_j^k = 1,\ x_j^k = 0,\ \forall\, j \notin I_k,\ w_j^k = 0,\ \forall\, j \in I_k \right\}, \tag{45}$$

where I_k is the set of indices defining the corresponding principal submatrix of A.

5 Computational Experience

We report some computational experience with the proposed enumerative algorithm in this section. All experiments were carried out using a personal computer with a 3.0 GHz Pentium IV processor and 2 GBytes of RAM memory. We used GAMS [29] to implement the most advanced version of the enumerative method, in which at each node t, a stationary point for NLP_2 was computed from a stationary point of $\text{NLP}_1(t)$ according to the procedure discussed in Section 4. Furthermore, stationary points for NLP_1 and NLP_2 (when necessary) were computed by using the solver MINOS [30].

In the first set of experiments, we solved the problems described in [24] of dimensions $n = 4, \dots, 7$, and for a given set of eigenvalues $\Lambda = \{\lambda_1, \dots, \lambda_p\}$, with $p = n+3, \dots, n+9$. The upper and lower bounds θ_2 and θ_1 for the entries of A were respectively set as $n \times \max_{k\in\mathbb{N}_p} |\lambda_k|$ and $-\theta_2$, and the values of Λ were determined from a uniform distribution on $[-1, 1]$.

We solved 10^3 test problems for each combination of n and p. Table 1 presents the performance of the enumerative algorithm, where the following notation is used: MAX NODES and AVERAGE, are respectively, the maximum and average numbers of searched nodes in the enumerative algorithm, and % PE is the percentage of problems that required the use of the enumerative algorithm to determine a solution (i.e., IECP was not solved at Step 0 of the algorithm). The numerical results show that the stationary point found for NLP_1 at the root node solves IECP in many cases (see values of (100-PE)%). Furthermore, the proposed method enumerated a very small number of nodes for all the 28×10^3 test problems.

In order to provide a comparison with an alternative technique, we report computational results for applying GAMS-BARON [31] directly to the nonlinear program NLP_1. These results are displayed in Table 2. In this table, MAX NODES stands for the maximum number of nodes visited by BARON to solve a problem, % PB is the percentage of problems that were not solved at the preprocessing phase, and % OS indicates the percentage of test problems for which BARON was able to identify a solution to the IECP within the time limit of 3600 CPU seconds. The results clearly show that the proposed enumerative algorithm is more

n		p n+3	n+4	n+5	n+6	n+7	n+8	n+9
4	MAX NODES	12	45	98	39	34	87	68
	AVERAGE	1.1	1.3	1.5	1.6	1.7	2.3	2.6
	% PE	2.7	7.0	9.3	15.4	23.4	23.4	30.4
5	MAX NODES	15	8	8	8	8	12	7
	AVERAGE	1.1	1.1	1.1	1.1	1.1	1.1	1.1
	% PE	2.9	2.6	2.7	4.0	5.5	6.2	8.5
6	MAX NODES	8	8	8	8	9	8	8
	AVERAGE	1.0	1.0	1.1	1.1	1.1	1.1	1.1
	% PE	1.7	1.4	3.6	3.7	4.9	1.5	6.4
7	MAX NODES	8	8	8	8	8	8	8
	AVERAGE	1.1	1.1	1.1	1.1	1.1	1.1	1.1
	% PE	6.4	2.4	3.9	4.1	7.8	13.0	20.8

Table 1 Performance of the enumerative algorithm (with 1000 instances for each size).

efficient than BARON for the solution of these IECPs.

The numerical results show that this choice is quite promising at least for this set of

n		p n+3	n+4	n+5	n+6	n+7	n+8	n+9
4	MAX NODES	262	159	1180	16478	8634	15184	23228
	% PB	18.1	63.7	74.4	80.2	96.8	97.2	98.1
	%OS	100.0	100.0	100.0	100.0	100.0	89.3	68.2
5	MAX NODES	45	176	182	67	157	92	221
	% PB	24.8	32.6	43.2	44.8	58.6	69.5	47.8
	%OS	100	100	100	100	100	100	100
6	MAX NODES	0	1	146	243	33	49	145
	% PB	0.0	22.4	24.1	35.4	34.7	72.5	41.8
	%OS	100.0	100.0	100.0	100.0	100.0	100.0	100.0
7	MAX NODES	0	0	0	0	1	4	63
	% PB	0.0	0.0	0.0	0.0	1.1	1.0	25.7
	%OS	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 2 Performance of the solver BARON for solving NLP_1 (with 1000 instances for each size).

problems, as it leads to better results than (43). However, formula (44) can only be applied when $p \leq n(n+1)/2$. This is the reason for the absence of results in Table 3 for $n = 4$ and $p = n+7, n+8, n+9$. In general, of course, one should apply (44) whenever applicable as indicated in Section 4.2.

As suggested by one of the referees, we tried to solve ten problems with the relatively larger values of $n = 10$ and $p = 20$. The enumerative method successfully solved 8 of these problems at node 1 by finding a stationary point of NLP_1 that turned out to be a solution to IECP. For the two remaining problems, the stationary point found for NLP_1 did not solve IECP. Furthermore, the procedure described in Section 4.1 did not provide a stationary point for NLP_2. Due to the large number of the so-called superbasic variables (see [30] for an ex-

n		p						
		n+3	n+4	n+5	n+6	n+7	n+8	n+9
4	MAX NODES AVERAGE % PE	25 1.1 4.1	61 1.5 16.2	45 2.8 37.1	8 1.0 14.6	-	-	-
5	MAX NODES AVERAGE % PE	3 1.0 0.1	2 1.0 0.5	4 1.0 1.8	9 1.0 3.5	10 1.1 6.1	6 1.2 12.2	9 1.3 18.9
6	MAX NODES AVERAGE % PE	2 1.0 0.1	2 1.0 0.1	6 1.0 0.3	2 1.0 0.3	2 1.0 0.4	3 1.0 1.1	3 1.0 1.1
7	MAX NODES AVERAGE % PE	1 1.0 0.0	1 1.0 0.0	2 1.0 0.1	2 1.0 0.3	2 1.0 0.8	2 1.0 3.0	2 1.0 7.3

Table 3 Performance of the enumerative algorithm with the bounds defined by (44) (with 1000 instances for each size).

planation of the meaning of these variables), the solver MINOS was unable to compute a stationary point for Problem NLP$_2$ from scratch. So at least for this set of problems, the inability to solve IECP is not directly related to the enumerative algorithm, and to better assess the proposed algorithmic approach, a more effective local NLP solver needs to be incorporated in the enumerative method for dealing with such IECPs of larger dimensions.

For the second set of experiments, we considered ECPs with the matrix $A = [2^{i+j}]$. Hence, A is non-negative and symmetric, and for each n, the ECP has $2^n - 1$ complementary eigenvalues of the form

$$\lambda^I = \sum_{i \in I} 2^{2i}, \tag{46}$$

where $I = \{i_1, \ldots, i_l\}$ are subsets of $\mathbb{N}_n$ with $l \geqslant 1$ and $1 \leqslant i_1 \leqslant \ldots \leqslant i_l \leqslant n$ [7]. Furthermore, each one of these complementary eigenvalues is an ordinary eigenvalue of the corresponding principal submatrix A_{II}. Based on these ECPs, we considered IECPs with the complementary eigenvalues given by (46), and performed the following experiments:

- Experiment I:

  We only assumed that A is a non-negative matrix by setting $\theta_1 = 0$. The enumerative algorithm found a solution for the problem with $n = 2$ in 4 nodes, but was not able to solve the IECP for $n \geqslant 3$.

- Experiment II:

  Besides the assumption $\theta_1 = 0$, we fixed the diagonal elements of the matrices equal to the corresponding eigenvalues with the consequent reduction of eigenvalues to

$p - n$. The enumerative algorithm found a solution for the instance $n = 3$ of these test problems, for both the asymmetric and symmetric cases, while exploring 29 and 12 nodes, respectively. However, problems having $n = 4$ and $n = 5$ remained unsolved.

- Experiment III:

  We fixed $\theta_1 = 0$ and forced each of the complementary eigenvalues to be an ordinary eigenvalue of a principal submatrix of A, that is, we imposed the constraints (45) discussed in Section 4 within the formulations NLP$_1$ and NLP$_2$. The enumerative algorithm was able to find a solution at the root node for all the IECPs with $n \leqslant 5$, where a different matrix from the one given by (46) was computed when symmetry was not required.

These results clearly indicate the importance of incorporating additional valid restrictions within the IECP in practice.

The third experiment suggested by one of the referees comprises the solution of the IECP with $n = 4$ and the spectral data specified in Table 4. For this example, choice (44)

$\lambda_1 = 16.2823$	$\lambda_7 = 27.5767$	$\lambda_{13} = 67.4575$	$\lambda_{19} = 148.0000$
$\lambda_2 = 16.4149$	$\lambda_8 = 31.0162$	$\lambda_{14} = 89.4233$	$\lambda_{20} = 187.1730$
$\lambda_3 = 18.7114$	$\lambda_9 = 36.4681$	$\lambda_{15} = 90.0000$	$\lambda_{21} = 194.5836$
$\lambda_4 = 19.1341$	$\lambda_{10} = 39.1435$	$\lambda_{16} = 97.5010$	$\lambda_{22} = 216.2813$
$\lambda_5 = 22.6080$	$\lambda_{11} = 56.9700$	$\lambda_{17} = 117.3920$	$\lambda_{23} = 221.9223$
$\lambda_6 = 22.8635$	$\lambda_{12} = 67.4251$	$\lambda_{18} = 138.5319$	–

Table 4 Experiment with $n = 4$ and $p = 23$.

for the lower and upper bounds θ_1 and θ_2 cannot be used. Furthermore the use of $\delta = n$ in the choice (43) was not good as the enumerative algorithm was unable to terminate in 1000 nodes. The reason for this seems to be the fact that the solution matrix to be computed by the enumerative method has some large elements in magnitude. So, we increased δ to $3n$ in this case, and the enumerative algorithm was able to terminate at the root node with an ϵ-optimal solution where termination was triggered by the objective value of NLP2 falling below 10^{-6}. For the obtained solution, the maximum discrepancy in the complementarity constraints ($\max\limits_{i,k}\{w_i^k x_i^k\}$) was 4.44E-07 and the maximum absolute discrepancy in the structural constraints ($\max\limits_{i,k} |w_i^k - A_{i.}x^k + \lambda_k x_i^k|$ where $A_{i.}$ is the ith row of A) was 5.96E-04. Despite the reasonable accuracy of this solution, when we computed all the complementary eigenvalues using the parametric algorithm introduced in [16], we came to the conclusion that this matrix indeed has a different set of complementary eigenvalues than the ones given, suggesting that a more accurate solution needs to be determined with tighter tolerances. Unfortunately, our attempt to do so turned out to be futile since the software MINOS was unable to compute stationary points with smaller values of the tolerances. Hence, we can only claim that the attained solution satisfies the IECP constraints within the stated tolerances. In our experience, this type of behavior may occur when p is much larger than n. As stated in Section 4.3, further constraints relating the elements of the solution matrix A and the given complementary eigenvalues could be added in order to reduce the value of p and the relative difficulty of the underlying IECP. Another possible strategy to deal with such challenging cases is to design a hybrid enumerative algorithm for the IECP in the spirit of the one described in [16]. This is certainly an interesting area of future research that we intend to pursue.

6 Conclusions

In this paper, we have considered a challenging class of problems known as the Inverse Eigenvalue Complementarity Problem (IECP), and have analyzed two nonlinear formulations NLP_1 and NLP_2. Our results exhibited that a stationary point for the NLP_1 can, in many cases, provide a solution to the IECP. However, to guarantee finding a solution to IECP when one exists, an enumerative method was designed that solves the problem by finding a global minimum for the program NLP_2. Some additional techniques were introduced for setting appropriate lower and upper bounds for the entries of the solution matrix A and for computing the stationary points required at each iteration of the enumerative algorithm. Computational experiments were reported using certain standard problems from the literature, which demonstrated that the enumerative algorithm is quite effective in practice, considering the complexity of this problem. For some instances, imposing certain implied structural constraints for the matrix A were instrumental in solving the problem. For future research, the use of such valid constraints and of enhanced local search techniques for improving the computational efficiency of the enumerative algorithm are areas that deserve attention. Another interesting topic for future research is the solution of the IECP on other convex cones, e. g., second-order cones. Furthermore, it is of interest to explore the solution of other inverse problems that occur in different classes of applications, including control theory and structural analysis.

Acknowledgements This research is partially supported by the project PEst-OE/MAT/ UI0297/2011, and by the National Science Foundation under Grant No. CMMI-0969169.

References

1. Demmel, J.W.: Applied Numer. Linear Algebra. SIAM, Philadelphia (1997)
2. Golub, G.H., Loan, C.F.V.: Matrix Computations, 3rd ed. John Hopkins University Press, Baltimore (1996)
3. Wilkinson, J.H.: The Algebraic Eigenvalue Problem. Clarendon Press, Oxford (1965)
4. Chu, M., Golub, G.H.: Inverse Eigenvalue Problems: Theory, Algorithms, and Applications. Oxford Science Publications, Oxford (2012)
5. Seeger, A.: Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions. Linear Alg. Appl. **292**(13), 1–14 (1999)
6. Pinto da Costa, A., Martins, J., Figueiredo, I., Júdice, J.J.: The directional instability problem in systems with frictional contacts. Comput. Meth. Appl. Mech. Eng. **193**(35), 357–384 (2004)
7. Pinto da Costa, A., Seeger, A.: Cone-constrained eigenvalue problems: theory and algorithms. Comput. Optim. Appl. **45**, 25–57 (2010)
8. Cottle, R., Pang, J., Stone, R.: The Linear Complementarity Problem. Academic Press, New York (1992)
9. Adly, S., Seeger, A.: A nonsmooth algorithm for cone-constrained eigenvalue problems. Comput. Optim. Appl. **49**, 299–318 (2011)
10. Brás, C.P., Fukushima, M., Júdice, J.J., Rosa, S.: Variational inequality formulation of the asymmetric eigenvalue complementarity problem and its solution by means of gap functions. Pac. J. Optim. **8(2)**, 197–215 (2012)
11. Júdice, J.J., Sherali, H.D., Ribeiro, I.M.: The eigenvalue complementarity problem. Comput. Optim. Appl. **37**, 139–156 (2007)
12. Júdice, J.J., Raydan, M., Rosa, S.S., Santos, S.A.: On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm. Numer. Algorithms **44**, 391–407 (2008)

13. Júdice, J.J., Sherali, H.D., Ribeiro, I.M., Rosa, S.: On the asymmetric eigenvalue complementarity problem. Optim. Method Softw. **24**, 549–586 (2009)
14. Le Thi, H., Moeini, M., Pham Dinh, T., Júdice, J.J.: A DC programming approach for solving the symmetric eigenvalue complementarity problem. Comput. Optim. Appl. **51**, 1097–1117 (2012)
15. Niu, Y.S., Pham, T., Le Thi, H.A., Júdice, J.J.: Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem. Optim. Method Softw. (to appear)
16. Fernandes, L.M., Júdice, J.J., Sherali, H.D., Fukushima, M.: On the computation of all eigenvalues for the eigenvalue complementarity problem. Manuscript, Instituto de Telecomunicações, Coimbra, Portugal
17. Fernandes, L.M., Júdice, J.J., Sherali, H.D., Forjaz, M.A.: On an enumerative algorithm for solving eigenvalue complementarity problems. Comput. Optim. Appl. (to appear)
18. Seeger, A.: Quadratic eigenvalue problems under conic constraints. SIAM J Matrix Anal. Appl. **32**(3), 700–721 (2011)
19. Seeger, A., Torki, M.: On eigenvalues induced by a cone constraint. Linear Alg. Appl. **372**, 181 – 206 (2003)
20. Zhou, Y., Gowda, M.S.: On the finiteness of the cone spectrum of certain linear transformations on Euclidean Jordan algebras. Linear Alg. Appl. **431**(57), 772–782 (2009)
21. Queiroz, M., Júdice, J.J., Humes, C.: The symmetric eigenvalue complementarity problem. Math. Comput. **73**, 1849–1863 (2003)
22. Seeger, A., Vicente-Pérez, J.: On cardinality of Pareto spectra. Electron. J. Linear Algebra **22**, 758–766 (2011)
23. Gajardo, P., Seeger, A.: Reconstructing a matrix from a partial sampling of pareto eigenvalues. Comput. Optim. Appl. **51(3)**, 1119–1135 (2012)
24. Gajardo, P., Seeger, A.: Solving inverse cone-constrained eigenvalue problems. Numer. Math. pp. 1–23 (2012)
25. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming : Theory and Algorithms, 3rd ed. John Wiley & Sons, New York (2006)
26. Facchinei, F., Pang, J.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, New York (2003)
27. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edition. Springer, New York (2006)
28. Sherali, H.D., Tuncbilek, C.H.: A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. J. Glob. Optim. **2**, 101–112 (1992)
29. Brooke, A., Kendrick, D., Meeraus, A., Raman, R.: GAMS-a User's Guide. GAMS Development Corporation, Washington (1998)
30. Murtagh, B., Saunders, M., Murray, W., Gill, P., Raman, R., Kalvelagen, E.: MINOS-NLP. Systems Optimization Laboratory, Stanford University, Palo Alto, CA
31. Sahinidis, N.V., Tawarmalani, M.: BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs. User's Manual (2005)