To appear in *Optimization Methods & Software*
Vol. 00, No. 00, Month 20XX, 1–35

An Alternating Direction Method of Multipliers for the Eigenvalue Complementarity Problem

Joaquim Júdice[a], Masao Fukushima[b], Alfredo Iusem[c], J. Mario Martinez[d], Valentina Sessa[e*]

[a]*Instituto Telecomunicações, Coimbra, Portugal.* [b]*Faculty of Science and Technology, Nanzan University, Nagoya 466-8673, Japan.* [c]*Instituto de Matemática Pura e Aplicada (IMPA), Rio de Janeiro, Brazil.* [d]*Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, Campinas, SP, Brazil.* [e]*MINES ParisTech, PSL Research University, CMA Centre de Mathématiques Appliquées, Sophia Antipolis, France.*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

We introduce an Alternating Direction Method of Multipliers (ADMM) for finding a solution of the nonsymmetric Eigenvalue Complementarity Problem (EiCP). A simpler version of this method is proposed for the symmetric EiCP, that is, for the computation of a Stationary Point (SP) of a Standard Fractional Quadratic Program. The algorithm is also extended for the computation of an SP of a Standard Quadratic Program (StQP). Convergence analyses of these three versions of ADMM are presented. The main computational effort of ADMM is the solution of a Strictly Convex StQP, which can be efficiently solved by a Block Principal Pivoting algorithm. Furthermore, this algorithm provides a stopping criterion for ADMM that improves very much its efficacy to compute an accurate solution of the EiCP. Numerical results indicate that ADMM is in general very efficient for solving symmetric EiCPs in terms of the number of iterations and computational effort, but is less efficient for the solution of nonsymmetric EiCPs. However, ADMM is able to provide a good initial point for a fast second-order method, such as the so-called Semi-smooth Newton method. The resulting hybrid ADMM and SN algorithm seems to be quite efficient in practice for the solution of nonsymmetric EiCPs.

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

AMS Subject Classification: 90C30; 90C33; 90C26; 93B60

1. Introduction

The Eigenvalue Complementarity Problem (EiCP) consists of finding $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n \setminus \{0\}$ such that:

$$w = Ax - \lambda Bx \tag{1a}$$

$$x \geq 0, w \geq 0 \tag{1b}$$

$$x^\top w = 0, \tag{1c}$$

[a] This work is funded by FCT/MEC through national funds and when applicable co-funded by FEDER - PT2020 partnership agreement under the project UID/EEA/50008/2019.
[*] Corresponding author Email: sessa.valentina@gmail.com

where A and B are given matrices of order n and B is positive definite (PD). The real number λ is called a complementary eigenvalue and x is the complementary eigenvector associated to λ. We denote by EiCP(A, B) an EiCP with matrices A and B. An EiCP is symmetric if both the matrices in its definition are symmetric (B is SPD). Otherwise, EiCP is said to be nonsymmetric.

Since the problem is homogeneous in x, it is sufficient to add the constraint

$$e^\top x = 1, \tag{1d}$$

with e being a vector of ones, for guaranteeing that x is a nonzero vector in a solution of EiCP. Furthermore, it follows from the complementarity condition (1c) that λ is given by

$$\lambda(x) = \frac{x^\top Ax}{x^\top Bx}, \tag{2}$$

in any solution (x, λ) of EiCP.

EiCP was introduced in [32] as an extension of the well-known Eigenvalue Problem [16] and has deserved great attention during the past several years. The problem has found applications in engineering [17, 28] and interest in the Spectral Theory of Graphs [14, 33]. Some generalizations of EiCP have been introduced in recent years [2, 7, 9, 15, 19, 30, 37, 39–41] and many interesting theoretical results have been presented in the past [6, 21, 29, 31, 32, 34–36]. Among them, it has been shown that EiCP always has a solution, as it is equivalent to a Variational Inequality (VI) Problem with a continuous function on the ordinary simplex [6]. Furthermore, the number of solutions of an EiCP can be exponential on the dimension n of EiCP [36]. Note that for the traditional Eigenvalue Problem this number is smaller than or equal to n [16].

A number of algorithms have been introduced for finding a solution of an EiCP. A Semi-smooth Newton (SN) method [3] has found great interest for solving nonsymmetric and symmetric EiCPs due to its fast convergence properties. However, global convergence is not guaranteed and the algorithm may fail for many instances. Splitting algorithms have been recommended for nonsymmetric and symmetric EiCPs with some interesting theoretical convergence properties [18]. As before, there is no guarantee that these algorithms converge in all the cases and the algorithms may fail for some instances. Descent algorithms for minimizing the so-called regularized gap function associated to the VI formulation of EiCP [1, 6] and DC algorithms [23, 26] have also been proposed for computing a solution of an EiCP. These algorithms face the same drawback, that is they may be unable to terminate successfully for some instances. An enumerative method has been introduced in [21] and is the unique algorithm that can always find a solution of EiCP in theory. The algorithm is efficient to solve EiCPs of small dimension [13, 21]. However, the algorithm may require too much work or may even be unable to compute a solution in a reasonable amount of CPU time for larger instances. This enumerative method can be combined with an SN method to improve its efficiency and can be extended to compute all the solutions of an EiCP of small dimension [13].

As discussed in [22, 31, 35], computing a solution of a symmetric EiCP is equivalent to finding a Stationary Point (SP) of a Standard Fractional Quadratic Program (StFQP), that is, an SP of a Fractional Quadratic function on the ordinary simplex. So, a local solver such as an active-set method [27] or an interior-point algorithm [27, 38] can be used to efficiently solve the symmetric EiCP. Despite being quite robust, these algorithms may be unable to solve large-scale symmetric EiCPs. Projected Gradient algorithms have been proposed for such EiCPs and showed to be very useful for some large and sparse instances particularly when the accuracy of the solution is not much at stake [8, 22]. However, these algorithms may fail to terminate when accuracy of the solution is on demand, particularly when at least one of the matrices of EiCP is ill-conditioned. As for the nonsymmetric EiCP, DC algorithms [23] and descent methods for minimizing the regularized gap function [1] may also be useful for computing a solution of the symmetric EiCP for large-scale instances. However, they share the same drawback of the projected-gradient methods.

In this paper, we introduce a new Alternating Direction Method of Multipliers (ADMM) for solving the nonsymmetric EiCP. The algorithm is an extension of the well-known ADMM that has been introduced for convex optimization [4, 11]. Specifically, by introducing an auxiliary variable y, we reformulate the EiCP as an optimization problem with the objective function $x^T Ax - x^T By$ and the additional coupling constraints $\lambda x - y = 0$ and $Ax - By - w = 0$, and then utilize the idea of ADMM. In addition to the presence of parameter λ, problem has some features that distinguish it from those problems which traditional ADMMs have been designed to solve. Particularly, the objective function is neither convex nor separable, that is, the second term of the objective function contains both variables x and y. Although there have been recent attempts to develop ADMMs that can handle certain nonconvex problems, separability of the objective function has usually been assumed in the existing ADMMs [5, 24].

The steps of ADMM are simpler when it is applied to the symmetric EiCP. Furthermore, an extension of the algorithm for the computation of an SP of a Standard Quadratic Program (StQP) is also introduced. In each iteration, primal feasibility and complementarity of the iterates (x, λ) are maintained, i.e., (x, λ) sastifies (1b), (1d) and (1c) (or (2)). The algorithm terminates when dual feasibility is satisfied, that is the vector w satisfies (1a) and (1b). In order to get such an objective, the algorithm employs an additional vector $y \in \mathbb{R}^n$ that should satisfy $\lambda x = y$ in any solution of the EiCP. As in the usual ADMM, an Augmented Lagrangian (AL) function is used to monitor the progress of the algorithm. In each iteration, this function is minimized alternatively for the original variables x_i on the simplex and for the unrestricted additional variables y_i. Finally, the dual variables introduced in the AL function are updated by the well-known procedure used in the so-called Augmented Lagrangian methods [27].

A partial convergence analysis of ADMM for solving the nonsymmetric EiCP is presented and shows that if the sequences of some of the primal and dual iterates converge, then the limit point $\bar{x}$ of the sequence of the primal iterates $\{x^k\}$ gives a solution $(\bar{x}, \bar{\lambda})$ of EiCP, where $\bar{\lambda} = \lambda(\bar{x})$ is given by (2). For the symmetric EiCP only the sequence $\{x^k\}$ has to converge for the same result to be true. Since this sequence is bounded, then it has an accumulation point $\bar{x}$ and we have been able to show that such a point gives a solution $(\bar{x}, \lambda(\bar{x}))$ of the symmetric EiCP provided two reasonable conditions hold. A similar convergence result is proven for the extension of ADMM for computing an SP of an StQP.

The main work of each iteration of ADMM relies on the solution of a Strictly Convex Standard Quadratic Program (SCStQP). A well-known Block Principal Pivoting (BPP) algorithm [20] is used to solve these SCStQPs and showed to be quite efficient in practice. Furthermore, a stopping criterion for ADMM is designed based on the BPP method that proves to be quite important to reduce the number of iterations for ADMM to get a sufficiently accurate solution for the EiCP. Despite the use of such an efficient subroutine, computational experiments reported in this paper indicates that ADMM is usually slow to compute a solution of a nonsymmetric EiCP particularly for large-scale instances. In our opinion, this is a consequence of the complexity of the Augmented Lagrangian Function. This function is much simpler for the symmetric EiCP. This makes ADMM simpler and more efficient for solving a symmetric EiCP.

The choice of the penalty parameter ρ has an important impact on the number of iterations of ADMM to terminate. Our experiments indicate that the number of iterations of ADMM usually reduce with a reduction of ρ. For many instances a value of $\rho = 20$ is enough for ADMM to work well. However, we found some instances where a much bigger value of ρ is required for the algorithm to terminate. On the other hand, there are instances where a quite small value of ρ, such as $\rho = 0.1$, improves dramatically the speed of ADMM to find an accurate solution. So, an efficient procedure to find an appropriate value of the penalty parameter for each instance is an important issue for future research.

A hybrid method has been introduced in [13] for solving the nonsymmetric EiCP. This method combines a slow enumerative algorithm with a fast Semi-smooth Newton (SN) method. In this paper, we propose to use the ADMM instead of the enumerative method in a similar hybrid scheme. The use of the new stopping criterion mentioned before with a larger tolerance usually provides a premature termination of ADMM with a good initial point for a fast SN method to work very well. Two hybrid methods are introduced in this paper that exploit this idea with two different SN algorithms. Computational experiments reported in this paper indicate that such hybrid approaches should be used in practice for the solution of the nonsymmetric EiCP. As stated before, ADMM has better convergence properties and seems to be much more efficient for solving the symmetric EiCP. Hence, the SN method does not seem to be required in this case and we recommend ADMM to be employed alone for the solution of the symmetric EiCP.

The structure of the paper is as follows. In Section 2, the new ADMM for the nonsymmetric EiCP is introduced together the details of its implementation and its convergence analysis. The extensions of this algorithm for the symmetric EiCP and for computing an SP of an StQP are discussed in Sections 3 and 4. Two Semi-smooth Newton methods and hybrid algorithms combining ADMM and SN methods are discussed in Section 5. Computational experience with the ADMM and hybrid methods is reported in Section 6. Finally, some conclusions are presented in the last section of the paper.

Notation: If x is a vector of dimension n, then $x \geq 0$ means that $x_i \geq 0$ for all $i = 1, ..., n$.

2. An ADMM for the nonsymmetric EiCP

Consider the EiCP(A, B) (1). If A is not a PD matrix, it is always possible to make this matrix to be PD by adding the matrix μB for some $\mu > 0$. Note that (1a) can be written as

$$w = (A + \mu B)x - (\lambda + \mu)Bx. \tag{3}$$

Then the following property holds:

Property 1 EiCP(A, B) has a solution (x, λ) if and only if EiCP$(A+\mu B, B)$ has a solution $(x, \lambda+\mu)$.

So, we can assume without loss of generality that A is a PD matrix in EiCP (1a)-(1d). Note that by (2) $\lambda > 0$ in any solution of EiCP.

Consider the following parametric quadratic problem (QP):

$$\begin{aligned} \mathrm{QP}_\lambda : \quad \min \quad & f_\lambda(x) = x^\top Ax - \lambda x^\top Bx && \text{(4a)} \\ \text{s.t.} \quad & Ax - \lambda Bx \geq 0, && \text{(4b)} \\ & x \in S, && \text{(4c)} \end{aligned}$$

where

$$S = \{x \in \mathbb{R}^n : e^\top x = 1, x \geq 0\}. \tag{5}$$

We recall that $\bar{x}$ is an SP of QP_λ if

$$\nabla f_\lambda(\bar{x})^\top (x - \bar{x}) \geq 0,\ \forall x \in P,$$

where P is the feasible convex set of QP_λ. Then the following theorem holds:

THEOREM 2.1 *x^* is an SP of QP_{λ^*} with $\lambda^* = \lambda(x^*)$ if and only if (x^*, λ^*) is a solution of EiCP.*

Proof. If x^* is an SP of QP_{λ^*}, then (x^*, λ^*) satisfies all the constraints of this QP and

$$(x^*)^\top A x^* - \lambda^*(x^*)^\top B x^* = 0.$$

So, (x^*, λ^*) is a solution of EiCP. On the other hand, if (x^*, λ^*) is a solution of EiCP then x^* is a global minimum for QP^*_λ, as x^* is a feasible solution satisfying

$$f_{\lambda^*}(x^*) = 0 \leq f_{\lambda^*}(x),$$

for any feasible solution x of QP_{λ^*}. So, x^* is an SP for QP_{λ^*}. ∎

In this section, we introduce an ADMM for finding a solution of the nonsymmetric EiCP. In order to explain the steps of ADMM, we consider the following Nonlinear Programming Problem (NLP) associated with EiCP:

$$\begin{aligned}
\min\quad & x^\top A x - x^\top B y && \text{(6a)}\\
\text{s.t.}\quad & x \in S && \text{(6b)}\\
& y \in \mathbb{R}^n && \text{(6c)}\\
& w \geq 0 && \text{(6d)}\\
& \lambda x - y = 0 && \text{(6e)}\\
& Ax - By - w = 0. && \text{(6f)}
\end{aligned}$$

In this NLP, x is the primal variable, y and w are the auxiliary variables and λ is a parameter. In the algorithm presented below, λ is updated at each iteration in such a way that (2) is satisfied, thereby maintaining the complementarity condition in EiCP (1).

Let $p \in \mathbb{R}^n$ and $q \in \mathbb{R}^n$ be the vectors of dual variables associated to the constraints (6e) and (6f). Consider the AL function:

$$\begin{aligned}
L_\rho(x, \lambda, y, w, p, q) = {} & x^\top A x - x^\top B y + p^\top(\lambda x - y) + q^\top(Ax - By - w)\\
& + \frac{\rho}{2}\Big(\|\lambda x - y\|^2 + \|Ax - By - w\|^2\Big). \qquad (7)
\end{aligned}$$

In each iteration k of ADMM, let $x^k \in S$, $y^k \in R^n$, $w^k \geq 0$ be the vectors of the current primal variables, and p^k, q^k be the vectors of the current dual variables of the AL function and $\lambda_k = \lambda(x^k)$ be the value of the parameter λ, where $\lambda(x)$ is given by (2). Then the vectors of the primal variables are updated by alternating minimization of the AL function and the parameter λ is computed by (2) with the updated vector x^{k+1}, i.e., x^{k+1}, λ_{k+1}, y^{k+1} and w^{k+1} are computed by:

$$\min_{x \in S} \quad L_\rho(x, \lambda_k, y^k, w^k, p^k, q^k)$$

$$\lambda_{k+1} = \lambda(x^{k+1})$$

$$\min_{y \in R^n} L_\rho(x^{k+1}, \lambda_{k+1}, y, w^k, p^k, q^k)$$

$$\min_{w \geq 0} L_\rho(x^{k+1}, \lambda_{k+1}, y^{k+1}, w, p^k, q^k)$$

respectively. Furthermore, the vectors of the dual variables are updated as in Augmented Lagrangian Methods. Algorithm 1 presents the steps of the ADMM.

Algorithm 1 : ADMM for nonsymmetric EiCP

▷ **Step 0: Initialization**

1: Set $k := 0$ and choose $x^0, y^0, w^0, p^0, q^0 \in \mathbb{R}^n$, λ_0 and $\rho > 0$.

▷ **Step 1: Iterations**

2: Compute x^{k+1} as the unique global minimum of the Strictly Convex Standard Quadratic Program (SCStQP):

$$\min_{x \in S} \ \left(\lambda_k p^k + A^\top q^k - By^k - \rho(\lambda_k y^k + A^\top(By^k + w^k))\right)^\top x + \frac{1}{2} x^\top (A + A^\top + \rho A^\top A + \rho \lambda_k^2 I)x \tag{8}$$

3: Compute

$$\lambda_{k+1} = \frac{(x^{k+1})^\top A x^{k+1}}{(x^{k+1})^\top B x^{k+1}} \tag{9}$$

Compute y^{k+1} as the unique global minimum of

$$\min_{y \in \mathbb{R}^n} \quad -c^\top y + \frac{1}{2} y^\top G y \tag{10}$$

with

$$c = p^k + B^\top q^k + B^\top x^{k+1} + \rho\Big(\lambda_{k+1} x^{k+1} + B^\top (A x^{k+1} - w^k)\Big) \tag{11}$$

and

$$G = \rho(I + B^\top B). \tag{12}$$

Compute w^{k+1} as the unique global minimum of

$$\min_{w \geq 0} \quad -\Big(q^k + \rho(A x^{k+1} - B y^{k+1})\Big)^\top w + \frac{1}{2} w^\top (\rho I) w \tag{13}$$

Update dual variables:

$$p^{k+1} = p^k + \rho(\lambda_{k+1} x^{k+1} - y^{k+1}) \tag{14}$$

$$q^{k+1} = q^k + \rho(A x^{k+1} - B y^{k+1} - w^{k+1}). \tag{15}$$

▷ **Step 2: Stopping criteria**

if (Criterion 1 OR Criterion 2) **then**

  terminate with (x^{k+1}, λ_{k+1}) being a solution of EiCP.

else

  set $k = k + 1$ and go to Step 1.

end if

(I) Solution of SCStQP

SCStQP (8) can be efficiently solved by a Block Principal Pivoting (BPP) method described in [20]. The KKT conditions for the unique global minimum of SCStQP (8) constitute the following Mixed LCP:

$$\begin{aligned} v &= h + Mz \\ z_i \geq 0, v_i &\geq 0, i = 1, ..., n \\ v_{n+1} &= 0, z_{n+1} \in \mathbb{R} \\ z^\top v &= 0, \end{aligned} \tag{16}$$

where z_{n+1} is the Lagrange multiplier associated to the equality constraint $e^\top x = 1$,

$$z = \begin{bmatrix} x \\ z_{n+1} \end{bmatrix}, \; M = \begin{bmatrix} A + A^\top + \rho A^\top A + \rho\lambda_k^2 I & -e \\ e^\top & 0 \end{bmatrix}$$

and

$$h = \begin{bmatrix} \lambda_k p^k + A^\top q^k - By^k - \rho(\lambda_k y^k + A^\top(By^k + w^k)) \\ -1 \end{bmatrix}.$$

So, M is a square matrix of order $(n+1)$ and z and h are $(n+1)$-vectors. Let $N = \{1, ..., n+1\}$. The BPP algorithm is used to solve this Mixed LCP by employing in each iteration the so-called Complementary Basic Solutions (CBS). In order to define a CBS, a partition $\{F, T\}$ of N is considered. The basic z- and v-variables are computed by:

$$\begin{aligned} M_{FF} z_F &= -h_F \\ v_T &= h_T + M_{TF} z_F. \end{aligned} \tag{17}$$

Furthermore, the nonbasic variables are given by:

$$z_T = 0, v_F = 0. \tag{18}$$

By assuming that $n+1$ always belongs to F, then v_{n+1} is always a nonbasic variable, which means that $e^\top x = 1$ for each CBS. Furthermore, there exists a CBS for each partition $\{F, T\}$ of N, since M_{FF} is always nonsingular, as a consequence of the matrix of the SCStQP being PD and e to be a vector of ones.

Given a CBS, then there are two possible cases:

(i) $z_i \geq 0$ for all $i \in F \setminus \{n+1\}$ and $v_T \geq 0$. Then this CBS is the unique solution of Mixed LCP (16) and $z = (z_F, 0)$ is the unique global optimal solution of SCStQP (8).

(ii) There exists at least an $i \in F \setminus \{n+1\}$ such that $z_i < 0$ or a $j \in T$ such that $v_j < 0$. Then, a new CBS is constructed with a new partition of N given by $\{F, T\}$, where F is updated by:

$$\begin{aligned} &F = F \setminus \{i \in \{F \setminus \{n+1\}\} : z_i < 0\} \cup \{i \in T : v_i \leq 0\} \\ &T = N \setminus F. \end{aligned} \tag{19}$$

Note that $v_i \leq 0$ is used in the update (19) instead of the more natural $v_i < 0$ by computational reasons. In fact, in practice the algorithm usually requires a smaller number of iterations to compute the unique solution of MLCP when the update (19) is employed. A new iteration of the BPP algorithm takes place with the new CBS. In practice, a positive tolerance ϵ (usually equal to 10^{-6}) is employed and the stopping criterion (SC) and the update (UPD) (19) are replaced respectively by:

$$\text{SC: } x_i \geq -\epsilon, \text{ for all } i \in F \setminus \{n+1\} \text{ and } v_j \geq -\epsilon, \text{ for all } j \in T.$$

$$\text{UPD: } F = F \setminus \{i \in \{F \setminus \{n+1\}\} : z_i < -\epsilon\} \cup \{i \in T : v_i \leq \epsilon\}.$$

The algorithm does not possess finite convergence in theory. As explained in [20], it is possible to combine this block principal pivoting version of BPP algorithm with a single principal pivoting version of BPP method that allows only one exchange of the set F in order to design a finitely convergent BPP algorithm. The so-called Murty's least-index pivoting method [25] can be used for this purpose. Given a CBS, an index $i \in N \setminus \{n+1\}$ is called an infeasibility if $i \in F$ and $z_i < 0$ or $i \in T$ and $v_i < 0$. The number of infeasibilities of a CBS is the number of such indices. We use this measure as a merit function that decides whether a block and fast update (19) or a single and slow update of only one exchange in the set F chosen by Murty's method is employed. However, in practice, Murty's method is never called and we skip the details of this finite convergence version of BPP algorithm.

An important point for the good efficiency of BPP algorithm in practice is the choice of the initial CBS. Our experiments suggest to use $F = N$ for the first iteration of ADMM. For the remaining iterations $k > 1$ of ADMM, the initial set F for BPP algorithm should be the set F corresponding to the CBS that is the unique optimal solution of SCStQP (8) of the previous iteration $k-1$. By using these initialization choices and due to the update (19), $n+1$ always belongs to F, as required.

(II) Other Computational Issues

(i) λ_{k+1} is computed as follows:

$$\begin{aligned} s &= Ax^{k+1} \\ r &= Bx^{k+1} \\ \lambda_{k+1} &= \frac{(x^{k+1})^\top s}{(x^{k+1})^\top r}. \end{aligned}$$

(ii) y^{k+1} is the unique optimal solution of the unrestricted strictly convex QP (10). Hence, y^{k+1} is the unique solution of the system of linear equations

$$Gy = c \tag{20}$$

where c and G are given by (11) and (12), respectively. The matrix G is SPD and there are very efficient direct and iterative algorithms for solving the linear system (20) for small and large n [16]. Furthermore, B is the identity matrix in many applications of eigenvalue problems. In this case, the matrix G of the system (20) is $2\rho I$ and y is given by $y = \frac{1}{2\rho}c$.

(iii) w^{k+1} is computed as the unique optimal solution of the strictly convex QP (13). Hence, w^{k+1} is computed by the following closed formula:

$$w^{k+1} = \max\left\{0, Ax^{k+1} - By^{k+1} + \frac{1}{\rho}q^k\right\}, \tag{21}$$

where max is taken componentwise. This formula is implemented as follows:

$$\begin{aligned} s &= Ax^{k+1} \\ u &= By^{k+1}. \end{aligned}$$

For $i = 1, ..., n$, set

$$w_i^{k+1} = 0$$
$$\gamma = s_i - u_i + \frac{1}{\rho} q_i^k.$$

If $\gamma \geq \epsilon$, set $w_i^{k+1} = \gamma$.

Note that s has been computed before in this k-th iteration.

(III) Convegence of ADMM for nonsymmetric EiCP

THEOREM 2.2 *If $\{(x^k, w^k, q^k)\}$ converges to (x^*, w^*, q^*) then (x^*, λ^*) with $\lambda^* = \lambda(x^*)$ is a solution of EiCP.*

Proof. Since the set S defined in (5) is compact and $x^k \in S$ for each k, then

$$x^* \in S. \tag{23}$$

Furthermore, the sequence $\{\lambda_k\}$ converges to

$$\lambda^* = \lambda(x^*) > 0, \tag{24}$$

where $\lambda(x^*)$ is given by (2).

Since y^{k+1} is a solution of the linear system (20), then y^{k+1} satisfies

$$\rho(\lambda_{k+1} x^{k+1} - y^{k+1}) + \rho B^\top (A x^{k+1} - B y^{k+1} - w^k) = -(B^\top x^{k+1} + p^k + B^\top q^k). \tag{25}$$

By using (25) and the formulas (14) and (15), we have

$$\begin{aligned} p^{k+1} &= p^k - \rho B^\top (A x^{k+1} - B y^{k+1} - w^k) - B^\top x^{k+1} - p^k - B^\top q^k \\ &= -B^\top (q^{k+1} - q^k) - \rho B^\top (w^{k+1} - w^k) - B^\top x^{k+1} - B^\top q^k. \end{aligned}$$

So,

$$p^{k+1} = -B^\top (x^{k+1} + q^{k+1}) - \rho B^\top (w^{k+1} - w^k). \tag{26}$$

Since the sequences $\{x^k\}$, $\{w^k\}$ and $\{q^k\}$ converge to x^*, w^* and q^*, respectively, then by (26), $\{p^k\}$ converges to $-B^\top (x^* + q^*)$. So, (14) implies that the sequence $\{y^k\}$ converges to

$$y^* = \lambda^* x^*. \tag{27}$$

Furthermore, by (21),

$$w^* = \max\{0, A x^* - B y^* + \frac{1}{\rho} q^*\} \geq 0, \tag{28}$$

and by (15),

$$A x^* - B y^* - w^* = 0. \tag{29}$$

Then, by (27) and (28), we have

$$Ax^* - \lambda^* Bx^* \geq 0. \tag{30}$$

On the other hand, by (28) and (29)

$$w^* = \max\left\{0, w^* + \frac{1}{\rho}q^*\right\},$$

which can be rewritten as

$$\min\left\{w^*, -\frac{1}{\rho}q^*\right\} = 0.$$

This implies

$$q^* \leq 0. \tag{31}$$

Next, we show that

$$(w^*)^\top q^* = 0. \tag{32}$$

Since $q^* \leq 0 \leq w^*$, we only have to prove that $w_i^* > 0$ implies $q_i^* = 0$ for each i. But $w_i^* > 0$ and (28) imply

$$w_i^* = \frac{1}{\rho}q_i^* + (Ax^* - By^*)_i.$$

Then, by (29)

$$q_i^* = -\rho(Ax^* - By^* - w^*)_i = 0.$$

Hence, by (27), (29) and (32), we have

$$((A - \lambda^* B)x^*)^\top q^* = 0. \tag{33}$$

As x^{k+1} is the unique optimal solution of the SCStQP (8), then x^{k+1} satisfies:

$$((A + A^\top)x^{k+1} - By^k + \lambda_k p^k + A^\top q^k + \rho(\lambda_k(\lambda_k x^{k+1} - y^k) + A^\top(Ax^{k+1} - By^k - w^k)))^\top(x - x^{k+1}) \geq 0, \tag{34}$$

for all $x \in S$. Since $\{x^k\}$, $\{q^k\}$ and $\{p^k\}$ converge to x^*, q^* and $-B^\top(x^* + q^*)$, respectively and (27) and (28) hold, then by passing (34) to the limit, we have

$$((A + A^\top)x^* - \lambda^*(B + B^\top)x^* + (A - \lambda^* B)^\top q^*)^\top(x - x^*) \geq 0, \tag{35}$$

for all $x \in S$. By (23), (24), (30), (31), (33) and (35), x^* is an SP of QP$_{\lambda^*}$ with $\lambda^* = \lambda(x^*)$. So, by Theorem 2.1, (x^*, λ^*) is a solution of EiCP. ■

Unfortunately, there is no guarantee that $\{(x^k, w^k, q^k)\}$ converges and the algorithm may be unable to compute a solution of the nonsymmetric EiCP. Despite the sequence $\{x^k\}$ being bounded, the remaining sequences $\{w^k\}$ and $\{q^k\}$ may be not bounded and an accumulation point of the sequence $\{(x^k, w^k, q^k)\}$ may not exist. In fact, the lack of separability in the objective function was the main obstacle to providing a complete convergence proof of the ADMM for NLP (6), as it prevents us from exploiting proof techniques used in the literature of ADMM. Finally, note that for the symmetric EiCP, we are able to prove in Section 3 that under two conditions any accumulation point x^* of the sequence $\{x^k\}$ gives a solution (x^*, λ^*), with $\lambda^* = \lambda(x^*)$.

(IV) Stopping Criteria for the nonsymmetric EiCP

Criterion 1:

Let F and T be the indices of basic and nonbasic variables associated to x^{k+1} and let λ_{k+1} be given by (9). Let

$$
\begin{aligned}
s &= Ax^{k+1} \\
r &= Bx^{k+1} \\
\sigma &= Ax^{k+1} - \lambda_{k+1}Bx^{k+1} = s - \lambda_{k+1}r.
\end{aligned}
$$

Then (x^{k+1}, λ_{k+1}) is a solution of EiCP if:

$$
\begin{aligned}
\sigma_i \geq -\epsilon,\ i \in T \\
-\epsilon \leq \sigma_i \leq \epsilon,\ i \in F
\end{aligned}
$$

Note: The vectors r and s have been computed before in this iteration k.

This criterion gives a solution of EiCP provided ϵ is a small positive tolerance. Computational experience to be reported later indicates that ϵ may be not too small for the algorithm to terminate with an accurate solution of EiCP. Our experience also shows that Criterion 1 has a dramatic effect on the efficiency of the algorithm. In fact, for many problems ADMM terminates fast (i.e., with a small number of iterations), whereas it is unable to terminate or is too slow if only Criterion 2 is employed. So, this Criterion 1 is an important contribution for ADMM to be useful in practice. Note that Criterion 1 is a consequence of the use of BPP algorithm to solve the required SCStQP in each iteration of ADMM.

Computer experiments with several test problems to be reported in this paper and elsewhere indicate that ADMM may be unable to compute a solution of the nonsymmetric EiCP for some instances. However, we can stop ADMM prematurely by using a bigger tolerance. As discussed in Section 5, this premature termination has an important effect on the efficiency of a hybrid algorithm for solving the nonsymmetric EiCP, which combines ADMM with a fast SN method.

Criterion 2:

This criterion follows from Theorem 2.2 and takes the following form:

$$\|x^{k+1} - x^k\| \leq \epsilon \text{ AND } \|w^{k+1} - w^k\| \leq \epsilon \text{ AND } \|q^{k+1} - q^k\| \leq \epsilon.$$

(V) Initial Point

The preprocessing technique described in [18] may be adapted to EiCP (1). For a canonical vector e^i to be a solution of EiCP (1), λ should satisfy

$$a_{ii} - \lambda b_{ii} = 0$$

as $x_i = 1 > 0$ implies $w_i = 0$. Hence,

$$\lambda = \frac{a_{ii}}{b_{ii}}.$$

Furthermore, for any $j \neq i$, we must have

$$w_j = a_{ji} - \frac{a_{ii}}{b_{ii}} b_{ji} \geq 0,$$

that is,

$$a_{ji} b_{ii} - a_{ii} b_{ji} \geq 0.$$

So, e^i is a solution of EiCP (1) if

$$\nu_i = \min\{a_{ji} b_{ii} - a_{ii} b_{ji} : j = 1, ..., n\} \geq 0. \tag{37}$$

So, the preprocessing technique looks for an $i \in \{1, ..., n\}$ satisfying (37). If such an i exists, then e^i is a solution of EiCP and ADMM is not required. Otherwise, $\nu_i < 0$ for all i and the initial point for ADMM is the vector $x^0 = e^\tau$, where

$$\tau = \text{argmax}\{\nu_i : i = 1, ..., n\}. \tag{38}$$

Alternatively, x^0 can be chosen as the barycenter $\frac{1}{n}e$ of the simplex (5). After computing x^0, the remaining components of the initial point are given by:

$$\lambda_0 = \frac{(x^0)^\top A x^0}{(x^0)^\top B x^0} \tag{39a}$$

$$y^0 = \lambda_0 x^0 \tag{39b}$$

$$w^0 = A x^0 - \lambda_0 B x^0 \tag{39c}$$

$$p^0 = q^0 = 0. \tag{39d}$$

By computing the auxiliary vectors defined before:

$$s = Ax^0$$
$$r = Bx^0,$$

then the computation of λ_0 and w_0 are as follows:

$$\lambda_0 = \frac{s^\top x^0}{r^\top x^0}$$
$$w^0 = s - \lambda_0 r.$$

3. An ADMM for the symmetric EiCP

In this case A and B are both symmetric matrices and B is SPD. As before, we can assume that A is also SPD. Consider the following parametric QP:

$$\begin{aligned} \text{QP}_\lambda: \quad & \min \quad f_\lambda(x) = \frac{1}{2}(x^\top Ax - \lambda x^\top Bx) \\ & \text{s.t.} \quad x \in S, \end{aligned} \tag{40}$$

where S is the simplex defined by (5). The following result can be proved:

Theorem 3.1 *x^* is an SP of QP_{λ^*} (40) with $\lambda^* = \lambda(x^*)$ if and only if (x^*, λ^*) is a solution of the symmetric EiCP.*

Proof. Let x^* be an SP of QP_{λ^*} with $\lambda^* = \lambda(x^*)$. Then x^* satisfies the KKT conditions

$$Ax^* - \lambda^* Bx^* = w + \theta e \tag{41a}$$
$$x^* \geq 0, \ w \geq 0 \tag{41b}$$
$$e^\top x^* = 1 \tag{41c}$$
$$(x^*)^\top w = 0, \tag{41d}$$

where $\theta \in \mathbb{R}$ and $w \in \mathbb{R}^n$ are the Lagrangian multipliers associated to the constraints $e^\top x = 1$ and $x \geq 0$, respectively. Hence, by (41a) and (41c) and the definition of λ^*, we have

$$0 = (x^*)^\top w + \theta.$$

Then (41d) implies $\theta = 0$. So, (x^*, λ^*) is a solution of EiCP. Conversely, if (x^*, λ^*) is a solution of EiCP, then x^* satisfies the KKT conditions (41) with $\theta = 0$. Then x^* is an SP for QP_{λ^*}. ∎

By Theorem 2.1, a solution of the symmetric EiCP can be computed by applying ADMM to an NLP obtained from (6) by replacing A and B by $\frac{1}{2}A$ and $\frac{1}{2}B$, respectively, and removing the constraint (6f) and the vector w. The AL function is

$$L_\rho(x, \lambda, y, p) = \frac{1}{2}x^\top Ax - \frac{1}{2}x^\top By + p^\top(\lambda x - y) + \frac{\rho}{2}\|\lambda x - y\|^2. \tag{42}$$

So, ADMM for the symmetric EiCP takes the following form:

Algorithm 2 : ADMM for symmetric EiCP

▷ **Step 0: Initialization**

Set $k := 0$ and choose $x^0, y^0, p^0 \in \mathbb{R}^n$, $\lambda_0 > 0$ and $\rho > 0$.

▷ **Step 1: Iterations**

Compute x^{k+1} as the unique global minimum of the SCStQP:

$$\min_{x \in S} \left(\lambda_k p^k - \frac{1}{2} B y^k - \rho \lambda_k y^k\right)^\top x + \frac{1}{2} x^\top (A + \rho \lambda_k^2 I) x \tag{43}$$

Compute $\lambda_{k+1} = \dfrac{(x^{k+1})^\top A x^{k+1}}{(x^{k+1})^\top B x^{k+1}}$.

Compute y^{k+1} by

$$y^{k+1} = \lambda_{k+1} x^{k+1} + \frac{1}{\rho}(p^k + \frac{1}{2} B x^{k+1}). \tag{44}$$

Update dual variables:

$$p^{k+1} = p^k + \rho(\lambda_{k+1} x^{k+1} - y^{k+1}).$$

▷ **Step 2: Stopping criteria**

if (Criterion 1 OR Criterion 2) **then**

  terminate with (x^{k+1}, λ_{k+1}) being a solution of EiCP.

else

  set $k = k + 1$ and go to Step 1.

end if

(I) Computational Issues

(i) The main effort of each iteration is the solution of an SCStQP (43), which is efficiently solved by BPP algorithm.

(ii) Contrary to the nonsymmetric case, the linear system of equations (20) is not required.

(iii) Since B is symmetric, (26) can be simplified as

$$p^{k+1} = -\frac{1}{2} B x^{k+1}. \tag{45}$$

So, instructions 4: and 5: should be implemented as follows:

$$
\begin{aligned}
r &= Bx^{k+1} \\
p^{k+1} &= -\frac{1}{2}r \\
y^{k+1} &= \lambda_{k+1}x^{k+1} + \frac{1}{\rho}(p^k - p^{k+1}).
\end{aligned}
$$

(II) Convergence of ADMM for the symmetric EiCP

By using a proof similar to and much simpler than the proof of Theorem 2.2, it is possible to show that if the sequence $\{x^k\}$ of iterates converges to x^*, then $(x^*, \lambda(x^*))$ is a solution of EiCP. Since this sequence is bounded then it has an accumulation point. The next theorem shows that such an accumulation point gives a solution of the symmetric EiCP under some conditions.

THEOREM 3.2 *Let x^* be an accumulation point of $\{x^k\}$ and $\{x^k\}_{k\in K}$, with $K \subseteq \{0, 1, ...\}$ be a subsequence of $\{x^k\}$ converging to x^*. If the conditions*

$$\lim_{k\in K} \|x^k - x^{k+1}\| = 0 \tag{46}$$

and

$$\lim_{k\in K} \|y^k - y^{k+1}\| = 0 \tag{47}$$

hold, then (x^*, λ^*) with $\lambda^* = \lambda(x^*)$ is a solution of the symmetric EiCP.

Proof. Note that x^* exists as the sequence $\{x^k\}$ is bounded. Since

$$\lim_{k\in K} x^k = x^* \tag{48}$$

then, by (46), we have

$$\lim_{k\in K} x^{k+1} = x^*. \tag{49}$$

Furthermore, by (48) and (49)

$$\lim_{k\in K} \lambda_k = \lim_{k\in K} \lambda_{k+1} = \frac{x^{*T}Ax^*}{x^{*T}Bx^*} = \lambda^*. \tag{50}$$

By (45),

$$\lim_{k\in K} p^k = \lim_{k\in K} p^{k+1} = -\frac{1}{2}Bx^*.$$

Now,

$$\lim_{k\in K} y^{k+1} = \lim_{k\in K} (\lambda_{k+1}x^{k+1} + \frac{1}{\rho}(p^k + \frac{1}{2}Bx^{k+1})) = \lambda^* x^*.$$

Hence, by condition (46),

$$\lim_{k\in K} y^k = \lambda^* x^* \tag{51}$$

Finally, as x^{k+1} is an SP of the SCStQP (43), then

$$(\lambda_k p^k - \frac{1}{2}By^k - \rho(x^{k+1} - y^k) + Ax^{k+1})^\top (x - x^{k+1}) \geq 0 \tag{52}$$

for all $x \in S$. By taking limits in (52) for $k \in K$, we have

$$(Ax^* - \lambda^* Bx^*)^\top (x - x^*) \geq 0$$

for all $x \in S$. Hence x^* is an SP of QP^*_λ (40) with λ^* given by (50). So, by Theorem 3.1, (x^*, λ^*) is a solution of the symmetric EiCP. ■

It is important to add that the assumptions (46) and (47) are less demanding than the hypothesis of the whole sequence $\{x^k\}$ to converge. Furthermore, computational experiments with symmetric EiCPs from different sources reported in Section 6.2 and elsewhere indicate that ADMM usually converges to a solution of the EiCP.

(III) Stopping Criteria

Criterion 1: Similar to the nonsymmetric case.

Criterion 2: $\|x^{k+1} - x^k\| \leq \epsilon$.

4. Computing a Stationary Point of a Standard Quadratic Programming Problem

Consider a Standard Quadratic Program (StQP)

$$\begin{aligned} \min \quad & c^\top x + \frac{1}{2}x^\top Qx \\ \text{s.t.} \quad & x \in S, \end{aligned} \tag{53}$$

where Q is a given symmetric matrix of order n, c is a given n-vector and S is defined by (5). If Q is an SPD matrix, then StQP is strictly convex (SCStQP) and has a unique global minimum. This point is the unique SP of StQP and the unique solution of Mixed LCP (16) with

$$M = \begin{bmatrix} Q & -e \\ e^\top & 0 \end{bmatrix} \text{ and } h = \begin{bmatrix} c \\ -1 \end{bmatrix}.$$

So, this SCStQP can be efficiently solved by the BPP algorithm discussed in Section 2. Now, consider the case where Q is not an SPD matrix. Then there exists an SPD matrix A (not unique) such that

$$Q = A - B.$$

Note that, contrary to the decomposition used in DC algorithms, B does not need to be symmetric positive semi-definite. However, B is a symmetric matrix, since Q and A are both symmetric. By introducing the additional vector $y \in \mathbb{R}^n$ then StQP can be written as follows:

$$\begin{aligned} \min \quad & c^\top x + \frac{1}{2}x^\top Ax - \frac{1}{2}x^\top By \\ \text{s.t.} \quad & x - y = 0 \\ & x \in S \\ & y \in \mathbb{R}^n, \end{aligned} \tag{54}$$

where S is the ordinary simplex given by (5). It is now easy to extend ADMM for computing an SP of StQP. The AL function takes the form:

$$L_\rho(x, y, p) = c^\top x + \frac{1}{2}x^\top Ax - \frac{1}{2}x^\top By + p^\top(x - y) + \frac{\rho}{2}\|x - y\|^2,$$

where $p \in \mathbb{R}^n$ is the dual vector associated to the constraint $x - y = 0$. Algorithm 3 is a simpler version of Algorithm 2 and its steps are presented below:

Algorithm 3 : ADMM for StQP

▷ **Step** 0**: Initialization**

1: Set $k = 0$ and choose y^0, $p^0 \in \mathbb{R}^n$ (usually $p^0 = y^0 = 0$), $\rho > 0$.

▷ **Step** 1**: Iterations**

2: Compute x^{k+1} by solving the following QP

$$\min_{x \in S} \quad \left(c - \frac{1}{2}By^k + p^k - \rho y^k\right)^\top x + \frac{1}{2}x^\top(A + \rho I)x. \tag{55}$$

3: Compute $y^{k+1} = x^{k+1} + \frac{1}{\rho}\left(p^k + \frac{1}{2}Bx^{k+1}\right)$.

4: Compute $p^{k+1} = p^k + \rho\left(x^{k+1} - y^{k+1}\right)$.

▷ **Step** 2**: Stopping criteria**

5: **if** (Criterion 1) OR (Criterion 2 **then**

6: terminate with x^{k+1} being a solution of StQP (53).

7: **else**

8: set $k = k + 1$ and go to Step 1.

9: **end if**

(I) Convergence of ADMM for finding an SP of an StQP

THEOREM 4.1 *Let x^* be an accumulation point of $\{x^k\}$ and $\{x^k\}_{k\in K}$, with $K \subseteq \{0, 1, ...\}$ be a subsequence of $\{x^k\}$ converging to x^*. If the conditions* (46) *and* (47) *hold, then x^* is an SP of StQP* (53).

Proof. The proof is similar to the proof of Theorem 3.2. ■

(II) Computational Issues

(i) Matrix B does not need to be stored. Furthermore, it is not used in the instructions of Algorithm 3 as we use

$$\begin{aligned} By^k &= Ay^k - Qy^k \\ Bx^{k+1} &= Ax^{k+1} - Qx^{k+1}. \end{aligned}$$

(ii) Instructions 3: and 4: should be implemented as follows:

$$\begin{aligned} r &= Ax^{k+1} \\ \tau &= Qx^{k+1} \\ p^{k+1} &= \frac{1}{2}(\tau - r) \\ y^{k+1} &= x^{k+1} + \frac{1}{\rho}(p^k - p^{k+1}). \end{aligned} \tag{56}$$

(iii) As before, BPP algorithm should be used to solve the SCStQP (55) in each iteration of Algorithm 3.

(iv) In practice, the choice of the SPD matrix A plays an important role. This choice is not unique and should be done by exploiting the structure and even the sparsity of the matrix Q of the quadratic function. In this paper, we only report experiments on the use of ADMM for solving the nonsymmetric and symmetric EiCP. The performance of ADMM for the computation of an SP of an StQP in practice will be reported in a future paper.

(v) Algorithm 3 can be extended to solve Quadratic Programs with Bounds (BQP) and Continous Knapsack Quadratic Problems (CKQP), as the BPP algorithm is very efficient to solve the required strictly convex quadratic programs [20].

(III) Stopping Criteria

Criterion 1:

The KKT conditions for StQP take the form:

$$\begin{aligned} c + Qx &= t + \mu e \\ t \geq 0&,\ \mu \in \mathbb{R} \\ x^\top t &= 0 \\ x &\in S, \end{aligned}$$

where S is the feasible set of StQP given by (5). So, in order to verify if x^{k+1} is an SP of StQP (53), we consider the following LP:

$$\begin{aligned} \min \quad & (x^{k+1})^\top t \\ \text{s.t.} \quad & t + \mu e = c + Qx^{k+1} \\ & t \geq 0, \ \mu \in \mathbb{R}. \end{aligned}$$

An optimal solution (μ, z) for this LP is computed as follows:

(i) $u = c + Qx^{k+1} = c + \tau$, where τ is computed in 3: of Algorithm 3 (see (56)).

(ii)

$$\mu = \min\{u_i : i = 1, \ldots, n\}. \tag{57}$$

(iii)

$$t = u - \mu e. \tag{58}$$

Then $t \geq 0$ and x^{k+1} is an SP of StQP (53) if $t^\top x^{k+1} \leq \epsilon$. So, this criterion should be implemented as follows:

Let F and T be the sets of the indices of the basic and nonbasic variables associated to x^{k+1} and let t be the vector given by (58), where μ is computed by (57). Since $x_T^{k+1} = 0$, x^{k+1} is an SP of StQP if

$$t_i \leq \epsilon, \ \text{for all } i \in F.$$

Criterion 2: $\|x^{k+1} - x^k\| \leq \epsilon$.

5. SN methods and a hybrid algorithm for EiCP

(I) Simple SN method

Consider EiCP as the following Mixed Nonlinear Complementarity Problem (Mixed NCP):

$$Ax - \lambda Bx - w = 0 \tag{59a}$$
$$e^\top x - 1 = 0 \tag{59b}$$
$$w_i \geq 0, \ x_i \geq 0, w_i x_i = 0, \ i = 1, \ldots, n, \tag{59c}$$
$$\lambda \in \mathbb{R}. \tag{59d}$$

By considering the Fischer-Burmeister (FB) function $\varphi_{FB}(a, b) = a + b - \sqrt{a^2 + b^2}$ with $a, b \in \mathbb{R}$, we can transform conditions (59c) into

$$\Phi(x, w) = \begin{bmatrix} \varphi_{\text{FB}}(x_1, w_1) \\ \vdots \\ \varphi_{\text{FB}}(x_n, w_n) \end{bmatrix} = 0.$$

Then Mixed NCP (59) is equivalent to finding a zero of the following function:

$$\Psi(x, w, \lambda) = \begin{bmatrix} Ax - \lambda Bx - w \\ e^\top x - 1 \\ \Phi(x, w) \end{bmatrix}. \tag{60}$$

Let $(\bar{x}, \bar{w}, \bar{\lambda})$ be the current iterate. If

$$\max\{\|A\bar{x} - \bar{\lambda}B\bar{x} - \bar{w}\|, |e^\top \bar{x} - 1|, \|\Phi(\bar{x}, \bar{w})\|\} < \epsilon, \tag{61}$$

then SN method terminates with $(\bar{x}, \bar{\lambda})$ being a solution of EiCP. Otherwise, a new direction is computed by considering the following system

$$Jd = -\zeta, \tag{62}$$

where $\zeta = \Psi(\bar{x}, \bar{w}, \bar{\lambda})$, $d = [d_x, d_w, d_\lambda]$ and J is given by

$$J(\bar{x}, \bar{w}, \bar{\lambda}) = \begin{bmatrix} A - \bar{\lambda}B & -I & -B\bar{x} \\ e^\top & 0 & 0 \\ V & Z & 0 \end{bmatrix}, \tag{63}$$

where V and Z are diagonal matrices with the following diagonal elements:

$$(V_{ii}, Z_{ii}) = \begin{cases} \left(1 - \dfrac{\bar{x}_i}{\sqrt{\bar{x}_i^2 + \bar{w}_i^2}}, 1 - \dfrac{\bar{w}_i}{\sqrt{\bar{x}_i^2 + \bar{w}_i^2}}\right) & \text{if } (\bar{x}_i, \bar{w}_i) \neq 0 \\ \left(1 - \widetilde{\xi}_i, 1 - \widetilde{\eta}_i\right) & \text{if } (\bar{x}_i, \bar{w}_i) = 0 \end{cases} \quad \forall i = 1, \ldots, n, \tag{64}$$

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

Now there are two cases:

(i) Matrix J is nonsingular and the search direction d is computed by solving the system (62).

(ii) Matrix J is singular and d is computed by finding the minimum norm solution of the following LSQ:

$$\min \|Jd + \zeta\|^2. \tag{65}$$

or by the Levenberg-Marquardt (LM) formula

$$\min \left\| \begin{bmatrix} J \\ \beta I \end{bmatrix} d + \begin{bmatrix} \zeta \\ 0 \end{bmatrix} \right\|^2. \tag{66}$$

where β is a real parameter and I is the identity matrix. Note that the LM formula (66) reduces to (65) if $\beta = 0$. Typically, the parameter β is updated at every iteration in such a way that it tends to zero as the algorithm converges to a solution.

Then a new iterate is computed by:

$$\tilde{x} = \bar{x} + d_x,\ \tilde{w} = \bar{w} + d_w,\ \text{and}\ \tilde{\lambda} = \bar{\lambda} + d_\lambda, \tag{67}$$

and a new iteration is performed with $\bar{x} = \tilde{x}$, $\bar{\lambda} = \tilde{\lambda}$ and $\bar{w} = \tilde{w}$.

The iterative algorithms based on (62) and (67) may also be referred to as Gauss-Newton method and modified Gauss-Newton method, respectively. Notice that the Fischer-Burmeister function has the property called semismoothness, which means that the function Ψ inherits the same property. Local superlinear or quadratic convergence of the semismooth Gauss-Newton (or modified Gauss-Newton) method can be established under suitable assumptions, including the nonsingularity of the Generalized Jacobian at the solution [12, Theo. 7.5.11]. Although these methods do not possess a global convergence property, we may expect that the hybridization with ADMM works effectively, since the latter algorithm used in the first stage will provide a good starting point for the (modified) Gauss-Newton method used in the second stage. Such a hybrid algorithm will be discussed later in this section and its efficiency and efficacy in practice will be reported in the next section.

(II) SN method with line search

Another form of trying to overcome the drawback of a singular Generalized Jacobian is to use the natural merit function associated to system (60), together with a line search procedure. Define the merit function associated to (60) as follows:

$$\psi(x, w, \lambda) = \frac{1}{2}\|\Psi(x, w, \lambda)\|^2. \tag{68}$$

Note that the above function is continuously differentiable, since the squared FB function is continuously differentiable. A simple version of a globally convergent SN method may be described as follows:

SN method with line search:

Step 0: Let $\epsilon_1, \epsilon_2 > 0$ be small enough, and β, δ, m, η, γ be constants such that $0 < \beta < 1$, $\delta > 0$, $m > 1$, $\eta > 0$, $0 < \gamma < 1$ (in practice, we use $\beta = 0.5$, $\delta \in (10^{-8}, 10^{-6})$, $m \in (2.0, 2.5)$, $\eta = 10^{-4}$, $\gamma = 0.5$). Choose a starting point (x^0, w^0, λ_0) and set $k := 0$.

Step 1: If $\psi(x^k, w^k, \lambda_k) < \epsilon_1$, terminate with a solution of Mixed NCP (59). If $\|\nabla\psi(x^k, w^k, \lambda_k)\| < \epsilon_2$, then terminate with an SP of the natural merit function (68) on R^{2n+1}.

Step 2: If $J(x^k, w^k, \lambda_k)$ given by (63) is singular, then go to Step 4. Otherwise, solve (62) to find the Newton direction (d_x, d_w, d_λ). If

$$\psi(x^k + d_x, w^k + d_w, \lambda_k + d_\lambda) \le \beta\psi(x^k, w^k, \lambda_k), \tag{69}$$

then set $(x^{k+1}, w^{k+1}, \lambda_{k+1}) = (x^k + d_x, w^k + d_w, \lambda_k + d_\lambda)$, $k := k + 1$ and go to Step 1. If (69) does not hold, then go to Step 3.

Step 3: If the descent condition

$$\begin{bmatrix} \nabla_x \psi(x^k, w^k, \lambda_k) \\ \nabla_w \psi(x^k, w^k, \lambda_k) \\ \nabla_\lambda \psi(x^k, w^k, \lambda_k) \end{bmatrix}^\top \begin{bmatrix} d_x \\ d_w \\ d_\lambda \end{bmatrix} \leq -\delta \left\| \begin{bmatrix} d_x \\ d_w \\ d_\lambda \end{bmatrix} \right\|^m$$

is satisfied, then go to Step 5. Otherwise, go to Step 4.

Step 4: Set $(d_x, d_w, d_\lambda) = -(\nabla_x \psi(x^k, w^k, \lambda_k), \nabla_w \psi(x^k, w^k, \lambda_k), \nabla_\lambda \psi(x^k, w^k, \lambda_k))$ and go to Step 5.

Step 5: Let α_k be the largest element in $\{1, \gamma, \gamma^2, \dots\}$ satisfying

$$\psi(x^k + \alpha_k d_x, w^k + \alpha_k d_w, \lambda_k + \alpha_k d_\lambda) \leq \psi(x^k, w^k, \lambda_k) + \eta \alpha_k \nabla \psi(x^k, w^k, \lambda_k)^\top (d_x, d_w, d_\lambda).$$

Set $(x^{k+1}, w^{k+1}, \lambda_{k+1}) = (x^k, w^k, \lambda_k) + \alpha_k (d_x, d_w, d_\lambda)$, $k := k + 1$ and go to Step 1.

Note that the components of the gradient $\nabla \psi(x, w, \lambda)$ at any point (x, w, λ) are given by

$$\begin{aligned} \nabla_x \psi(x, w, \lambda) &= (A - \lambda B)^\top (Ax - \lambda Bx - w) + \nabla_x \psi_{FB}(x, w) + (e^\top x - 1)e \\ \nabla_w \psi(x, w, \lambda) &= -(Ax - \lambda Bx - w) + \nabla_w \psi_{FB}(x, w) \\ \nabla_\lambda \psi(x, w, \lambda) &= -(Bx)^\top (Ax - \lambda Bx - w), \end{aligned}$$

where $\nabla_x \psi_{FB}(x, w)$ and $\nabla_w \psi_{FB}(x, w)$ represent the gradients of the squared Fischer-Burmeister function, i.e., $\psi_{FB}(x, w) = \frac{1}{2} \Phi(x, w)^\top \Phi(x, w)$. Specifically, $\nabla_x \psi_{FB}(x, w)$ and $\nabla_w \psi_{FB}(x, w)$ can be computed as

$$\begin{aligned} \nabla_x \psi_{FB}(x, w) &= \nabla_x \Phi(x, w) \Phi(x, w) = V \Phi(x, w) \\ \nabla_w \psi_{FB}(x, w) &= \nabla_w \Phi(x, w) \Phi(x, w) = Z \Phi(x, w), \end{aligned}$$

where V and Z are diagonal matrices whose diagonal elements are given by (64) with $(\bar{x}_i, \bar{w}_i)$ replaced by (x_i, w_i) for $i = 1, \dots, n$.

At each iteration of the algorithm, the Newton step is first checked. If it should not be accepted, i.e., either the generalized Jacobian is singular or (69) fails to hold, then the steepest descent direction is used to decrease the merit function ψ.

The algorithm stated above is an adaptation of the general globalized Newton-type method for solving nonsmooth systems $F(x) = 0$, where F is nonsmooth and F^2 is continously differentiable. The convergence properties of such an algorithm are well understood. Specifically, every accumulation point of the sequence generated by the algorithm is a stationary point of the function F^2 on R^{2n+1}. So, the algorithm may converge to a point that is not a solution of the system $F(x) = 0$. This is a drawback of the algorithm. On the positive side, under a suitable regularity condition, a stationary point of the function F^2 on R^{2n} is a solution of the system $F(x) = 0$ [12, pp. 752-753]. Furthermore, if the starting point is already sufficiently close to the solution, the Newton step is usually accepted, and the algorithm converges to the solution at least linearly, normally superlinearly, without using steepest descent steps. Furthermore, ADMM can provide such an initial point and we expect the SN method with such an initial point to be able to find a solution of EiCP. This procedure will be discussed next and its efficiency and efficacy in practice will be reported in the next section.

(III) Hybrid Algorithm

In this procedure, ADMM is firstly applied until Criterion 1 is satisfied with a relatively large tolerance ($\epsilon = 10^{-1}$). Then SN method (simple or with line search) is employed until the end with initial point given by:

$$x = x^{k+1},\ \lambda = \lambda_{k+1} \text{ and } w = \sigma$$

where $(x^{k+1}, \lambda_{k+1}, \sigma)$ satisfies Criterion 1 of ADMM.

6. Computational experiments on the solution of nonsymmetric and symmetric EiCPs

6.1 *EiCP Test Problems*

For Test Problems 1 and 2, we consider A as a nonsymmetric PD matrix of the form

$$A = C + \mu I, \tag{70}$$

where C is a randomly generated matrix with elements uniformly distributed in the interval $[-2, 10]$ and $\mu > |\min\{0, \theta\}|$, where θ is the smallest eigenvalue of $C + C^\top$. So, A is a nonsymmetric PD matrix. B is taken as the identity matrix in Test Problems 1, whereas in Test Problems 2, $B = P$, where P is a symmetric strictly diagonally dominant matrix with positive diagonal elements of the following form:

$$P_{i,i} = 10, \quad i = 1, \ldots, n, \tag{71a}$$
$$P_{i,j} = -1, \quad j = i+1, \ldots, i+4,\ i = 1, \ldots, n, \tag{71b}$$
$$P_{i,j} = -1, \quad j = i-1, \ldots, i-4,\ i = 1, \ldots, n. \tag{71c}$$

Hence, P is an SPD matrix. Note that the solution of EiCPs of this form has been reported in other computational studies of EiCP (see for instance [18]).

Test Problems 3 are symmetric EiCPs, whose solution has also been reported in other computational studies of EiCP (see for instance [18]). For these problems B is the identity matrix and A is an SPD matrix from the Harwell-Boeing collection.

Test Problems 4 are instances of EiCPs discussed in the so-called Spectral Theory of Graphs [14]. So, $B = I$ and A is the symmetric adjacency matrix of some of the larger graphs of the DIMACS collection [10]. Since A is not an SPD matrix we use Property 1 and for each instance we solve EiCP$(A + \mu B, B)$ instead of EiCP(A, B), where μ is a positive number so that $A + \mu B$ is an SPD matrix. Furthermore, for each instance the complementary eigenvalue of EiCP(A, B) is equal to $\lambda - \mu$, where λ is the value computed by ADMM.

Finally, we generate Test Problems 5 and Test Problems 6 with

$$A = C + C^\top + D,$$

where C is defined as in Test Problems 1 and D is chosen so that A is an SPD matrix. Furthermore, $B = I + D$ in Test Problems 5 and $B = P + D$ in Test Problems 6.

Each problem is denoted by the name used in the collection and by the dimension n of the EiCP, that is the order of each matrix A and B, which is included in brackets after the corresponding notation. The instances of Test Problems 1, 2, 5 and 6 are indicated as RAND(n), and we considered $n = 50,\ 100,\ 250,\ 500,\ 750,$ and 1000. As the resulting EiCP Test Problems 3 – 6 are symmetric, we use Algorithm 2 for computing a solution.

6.2 *Numerical results*

In the tables, we use the following notation:

- ρ: value of ρ used for the results presented in the table.
- It: number of iterations required by ADMM.
- λ: computed complementary eigenvalue.
- compl $= x^\top w$, where x and w are the vectors computed by ADMM.
- dualfeas $= \min\{w_i : i = 1, ..., n\}$, where w_i are the components of the vector w.
- nitBPP: average number of iterations required by BPP algorithm.
- Crit: 1 or 2 depending on the criterion that is satisfied for the algorithm to terminate.
- cputime: computational time required by BPP algorithm.

We set the stopping tolerances as $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$ for Criteria 1 and 2, respectively. The maximum number of iterations allowed for this ADMM is $nit_{max} = 6000$. In case the algorithm is not able to terminate within nit_{max} iterations, we run again the algorithm with tolerance 10^{-3} for Criterion 1. We write a * if it terminates satisfying Criterion 1 with this tolerance and ** in case the algorithm fails, that is, it attains nit_{max} iterations without satisfying one of the two criteria. The initial point $(x^0, \lambda_0, w^0, p^0, q^0)$ was computed as discussed in Section 2 with x^0 being the barycenter of the simplex (5) and the remaining components computed by (39). We have also tried in our experiments to choose $x^0 = e^\tau$, where τ is given by (38) but the barycenter seems to be, in general, a better choice at least for these test problems.

Our experiments have shown that ρ does not need to be large for both the versions of ADMM to work well. Furthermore, the number of iterations of ADMM to compute a solution of EiCP usually decreases with a reduction of ρ. In general, $\rho = 20$ is a good choice for solving the test problems presented in this paper. For some test problems, a larger value of ρ is required for ADMM to find a solution to the EiCP (see results of Test Problems 5 and 6 in Tables 11 and 12, respectively). Finally, it is interesting to see that for Test Problems 4 associated to adjacency matrices of graphs, ADMM performs quite well using a very small value of the penalty parameter, $\rho = 0.1$ (see Table 10). So, the choice of ρ in practice is still an issue that deserves more attention in the future.

The numerical results concerning the performance of ADMM for solving nonsymmetric EiCP Test Problems 1 and 2 are displayed in Tables 1 and 2. For all these test problems, BPP algorithm is quite efficient as, in general, it requires very few iterations to terminate. This can be verified by noticing the very small values of nitBPP in all the tables containing numerical results of the performance of ADMM for solving nonsymmetric and symmetric EiCPs. Algorithm 1 was able to solve all instances, but the larger of Test Problems 1. Furthermore, whenever successful, the algorithm always terminates with Criterion 1 based on the BPP method and computes an accurate solution of EiCP, as it is confirmed by the values in columns compl and dualfeas. ADMM is fast (requires a small number of iterations) to compute a solution of Test Problems 2 and slow (requires many iterations) for Test Problems 1. For the larger Test Problems 1, ADMM either terminates using a larger tolerance 10^{-3} in Criterion 1 (see instance with $n = 500$) or is unable to terminate even with this tolerance (see the instances with $n = 750$ and $n = 1000$).

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLSyst	Crit	cputime
RAND(50)	1734	20	2.6784e+02	9.1458e-17	-3.2799e-06	1.0000	3468	1	3.3860e-01
RAND(100)	2431	20	4.9331e+02	1.1883e-16	-1.7007e-06	1.0000	4862	1	1.7983e+00
RAND(250)	4745	20	1.1593e+03	2.8257e-16	-5.7327e-07	1.0000	9490	1	1.6823e+01
RAND(500)*	5474	20	2.2109e+03	5.8443e-16	-2.4410e-06	1.0000	10948	1	1.1643e+02
RAND(750)**	6000								
RAND(1000)**	6000								

Table 1.: Performance of ADMM for nonsymmetric EiCP - Test Problems 1.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLsyst	Crit	cputime
RAND(50)	30	20	1.1995e+02	-8.5706e-16	-4.7148e-05	1.0000	60	1	7.8285e-03
RAND(100)	23	20	2.3437e+02	-9.1604e-16	-7.8750e-05	1.0000	46	1	1.6802e-02
RAND(250)	22	20	5.6756e+02	-6.2125e-16	-4.8593e-05	1.0000	44	1	1.4335e-01
RAND(500)	32	20	1.0938e+03	-3.9340e-16	-3.3674e-05	1.0625	66	1	8.5641e-01
RAND(750)	23	20	1.6227e+03	5.3751e-16	-1.9455e-05	1.0870	48	1	1.2356e+00
RAND(1000)	23	20	2.1392e+03	-1.0842e-15	-3.3962e-05	1.0870	48	1	2.5072e+00

Table 2.: Performance of ADMM for nonsymmetric EiCP - Test Problems 2.

We also solved the same instances by using the hybrid algorithm discussed in Section 5, that is, Algorithm 1 is used until Criterion 1 is satisfied with tolerance 10^{-1}, and one of the SN methods discussed in the same section is applied until the end. The initial point for these SN methods is given by $x^0 = x$, $\lambda_0 = \lambda$ and $w = s - \lambda r$, where x and λ satisfy Criterion 1 with $\epsilon = 10^{-1}$, and s and r are computed in instruction 2: and also in testing Criterion 1 of Algorithm 1. The results obtained with the use of the hybrid method involving the simple SN method employing formula (65) for the case of the singularity of the Generalized Jacobian and a tolerance $\epsilon = 10^{-6}$ are shown in Tables 3 and 5, where we indicated by niterSN the number of iterations required by the SN method. In Tables 4 and 6, we show the results obtained by the hybrid method involving the SN method with line search. Note that the performance of the two versions of the hybrid method is similar, as both the algorithms starting with the initial point given by Algorithm 1 terminate with the same solution of EiCP in the same number of iterations for almost all the instances. Furthermore, the simple SN method performs better than SN method with line search when their performances are not the same. It is important to add that both the versions of SN algorithm require a quite small number of iterations to compute a solution of EiCP. Moreover, for each one of the Test Problems 1 and 2, both the versions of SN method compute the same complementary eigenvalue. Finally, the use of the hybrid version of this ADMM enables us to solve all the instances of Test Problems 1 and 2 successfully.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	niterSN	nLSyst	cputime
RAND(50)	313	20	2.6784e+02	-1.5288e-09	-6.1292e-08	1.0000	5	631	7.2736e-02
RAND(100)	509	20	4.9331e+02	-4.7342e-10	-3.7027e-08	1.0000	5	1023	4.0445e-01
RAND(250)	459	20	1.1593e+03	-1.3030e-10	-1.4521e-08	1.0000	5	923	1.5872e+00
RAND(500)	195	20	2.2065e+03	-9.8314e-11	-6.6620e-09	1.0000	5	395	4.0818e+00
RAND(750)	244	20	3.2514e+03	-2.3514e-11	-4.4542e-09	1.0000	5	493	1.0962e+01
RAND(1000)	598	20	4.2970e+03	-3.5416e-12	-3.2795e-09	1.0000	5	1201	4.8279e+01

Table 3.: Performance of the hybrid algorithm (ADMM and simple SN method) for nonsymmetric EiCP - Test Problems 1.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	niterSN	nLsyst	cputime
RAND(50)	313	20	2.6784e+02	-5.9576e-11	-2.2001e-09	1.0000	4	630	1.1530e-01
RAND(100)	509	20	4.9331e+02	-3.9166e-09	-3.0629e-07	1.0000	5	1023	3.9718e-01
RAND(250)	459	20	1.1593e+03	-3.3948e-11	-6.4833e-09	1.0000	7	925	1.4483e+00
RAND(500)	195	20	2.2065e+03	-4.8058e-10	-1.2969e-07	1.0000	7	397	3.8187e+00
RAND(750)	244	20	3.2514e+03	-1.6654e-09	-6.7455e-07	1.0000	6	494	1.1054e+01
RAND(1000)	598	20	4.2970e+03	-3.9204e-10	-3.6280e-07	1.0000	6	1202	5.0520e+01

Table 4.: Performance of the hybrid algorithm (ADMM and SN method with line search) for nonsymmetric EiCP - Test Problems 1.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	niterSN	nLsyst	cputime
RAND(50)	19	20	1.1995e+02	-3.8004e-09	-3.6454e-08	1.0000	5	43	9.4728e-03
RAND(100)	16	20	2.3437e+02	-3.7087e-10	-1.1475e-08	1.0000	5	37	2.5187e-02
RAND(250)	18	20	5.6756e+02	-5.6973e-11	-5.4711e-09	1.0000	4	40	1.4660e-01
RAND(500)	24	20	1.0938e+03	-1.3353e-11	-6.3715e-09	1.0833	5	55	9.4552e-01
RAND(750)	16	20	1.6227e+03	-3.8841e-10	-4.1850e-09	1.1250	5	39	1.7517e+00
RAND(1000)	17	20	2.1392e+03	-9.2747e-11	-3.1189e-09	1.1176	5	41	3.8099e+00

Table 5.: Performance of the hybrid algorithm (ADMM and simple SN method) for nonsymmetric EiCP - Test Problems 2.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	niterSN	nLsyst	cputime
RAND(50)	19	20	1.1995e+02	-5.5153e-11	-1.2674e-09	1.0000	6	44	1.2819e-02
RAND(100)	16	20	2.3437e+02	-9.6312e-11	-3.8991e-09	1.0000	5	37	2.4578e-02
RAND(250)	18	20	5.6756e+02	-5.6973e-11	-5.4711e-09	1.0000	4	40	1.2264e-01
RAND(500)	24	20	1.0938e+03	-3.5900e-12	-1.6765e-09	1.0833	7	57	1.1556e+00
RAND(750)	16	20	1.6227e+03	-2.8663e-10	-3.3034e-09	1.1250	7	41	2.3558e+00
RAND(1000)	17	20	2.1392e+03	-3.4965e-11	-2.6801e-09	1.1176	7	43	4.8234e+00

Table 6.: Performance of the hybrid algorithm (ADMM and SN method with line search) for nonsymmetric EiCP - Test Problems 2.

Since each iteration of BPP algorithm requires the solution of exactly one linear system, a linear system of the form (20) has to be solved in each iteration of Algorithm 1 and at most a linear system is required for the SN methods in each iteration, then the total number of linear systems to be solved by the hybrid method is:

$$nLsyst = It \times (nitBPP + 1) + niterSN. \tag{72}$$

On the other hand, for the simple ADMM this number is:

$$nLsyst = It \times (nitBPP + 1). \tag{73}$$

These values are included in all the tables and clearly indicate the better performance of the hybrid method over the simple ADMM for solving the nonsymmetric EiCP. Note that for the symmetric EiCP, ADMM does not require the system (20) to be solved in this iteration. So, +1 is removed from the formula (72).

In order to have a better idea of the efficiency of ADMM, we also solve the same test problems by using the best of the splitting algorithms discussed in [18]. Note that EiCP (1) is equivalent to

$$w = (-\lambda)Bx - (-A)x \tag{74a}$$

$$x \geq 0, w \geq 0 \tag{74b}$$

$$x^\top w = 0. \tag{74c}$$

This is the formulation of EiCP that was used for the splitting algorithms described in [18]. Since A is PD then $-A$ is ND and Algorithm A1 in [18] is the best splitting method to be used. Furthermore, according to (74), if (λ, x) is the complementary pair computed by the splitting algorithm A1, then $(-\lambda, x)$ is a solution of EiCP (1).

For the nonsymmetric EiCPs, we report the performance of the splitting algorithm A1 in Tables 7 and 8. The initial point was chosen as the barycenter of the simplex (5) and λ_0 as in (39a). Algorithm A1 was not able to solve the larger three instances of both Test Problems 1 and 2 within the allowed number of iterations, which has been set equal to 6000. Furthermore, when successful, splitting algorithm A1 usually requires more iterations, linear systems and CPU time than the hybrid methods.

Problem	It	λ	compl	dualfeas	nitBPP	nLSyst	cputime
RAND(50)	210	6.1097e+01	3.3029e-15	-1.5530e-05	2.0952	440	1.3530e-02
RAND(100)	722	9.6815e+01	5.0415e-17	-2.0556e-05	2.1039	1519	6.6086e-02
RAND(250)	3238	1.5660e+02	-5.6984e-16	-1.2882e-04	2.1288	6893	4.9123e-01
RAND(500)**	6000						
RAND(750)**	6000						
RAND(1000)**	6000						

Table 7.: Performance of the splitting Algorithm A1 for nonsymmetric EiCP - Test Problems 1.

Problem	It	λ	compl	dualfeas	nitBPP	nLSyst	cputime
RAND(50)	272	6.4135e+00	1.6754e-15	-2.7045e-05	2.0625	561	1.4692e-02
RAND(100)	1061	9.4452e+00	1.0355e-15	-2.1442e-05	2.1244	2254	9.6551e-02
RAND(250)	1269	1.5834e+01	9.6361e-16	-1.3892e-04	2.1174	2687	2.3668e-01
RAND(500)**	6000						
RAND(750)**	6000						
RAND(1000)**	6000						

Table 8.: Performance of the splitting Algorithm A1 for nonsymmetric EiCP - Test Problems 2.

As a final conclusion of this study on these Test Problems 1 and 2 and other experiments with similar nonsymmetric EiCP test problems, we claim that ADMM performs well for solving small dimensional nonsymmetric EiCPs but seems to be slow in general particularly for larger dimensional problems. Despite this drawback, ADMM seems to be more efficient than the best splitting method discussed in [18] for the solution of the nonsymmetric EiCP. The good performance of ADMM is strongly related to the use of BPP algorithm that is employed as its main subroutine. A stopping criterion based on this BPP algorithm, that is Criterion 1, improves very much the speed of ADMM towards a solution of EiCP. Furthermore, the use of this criterion with a larger tolerance provides a good initial point for a fast local convergent algorithm such as a semi-smooth Newton method. Two versions of the SN method have been tested and seem to be quite efficient for solving the nonsymmetric EiCP when such an initial point is used.

Next, we report the performance of ADMM for the solution of the symmetric EiCP Test Problems 3 – 6. Algorithm 2 is quite efficient to solve all the symmetric EiCP Test Problems 3 and 4 when an appropriate value of ρ is chosen. In our opinion, the first reason for the better performance of Algorithm 2 over Algorithm 1 is the Lagrangian function (42) which is much simpler and contains a much smaller number of dual variables than the Lagrangian function (7) used by Algorithm 1. As for Algorithm 1, BPP algorithm for solving the required SCStQPs and Criterion 1 based on this algorithm are the two relevant reasons for the efficiency and efficacy of Algorithm 2. Algorithm 2 found some difficulties for solving larger instances of Test Problems 5 and 6. As mentioned before, Algorithm 2 needed to use a large value of ρ in order to terminate. Furthermore, for some of the larger instances the algorithm was forced to terminate with Criterion 2 and obtained a less accurate solution.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLsyst	Crit	cputime
BCSSTK02(66)	4	20	7.6063e+00	-6.3527e-22	-3.1244e-05	3.2500	13	1	9.2219e-03
BCSSTK04(132)	4	20	1.6954e+02	-2.6470e-23	-1.3028e-05	1.7500	7	1	2.2428e-02
BCSSTK05(153)	1	20	9.6424e+02	4.3013e-22	-4.5617e-06	2.0000	2	1	5.3069e-03
BCSSTK10(1086)	1	20	1.1064e+05	2.1043e-21	-1.8066e-05	1.0000	1	1	1.1771e-01
BCSSTK27(1224)	1	20	1.2345e+04	-8.4703e-22	-1.9964e-05	1.0000	1	1	1.3975e-01
s1rmq4m1(5489)	1	20	2.5729e+01	-9.3058e-25	-1.8185e-07	1.0000	1	1	5.9845e+00
s1rmt3m1(5489)	1	20	1.7386e+01	-1.0340e-24	-8.4734e-08	1.0000	1	1	5.5897e+00
s2rmq4m1(5489)	1	20	1.8304e+00	-1.4476e-24	-1.1396e-07	1.0000	1	1	5.4893e+00

Table 9.: Performance of ADMM for symmetric EiCP - Test Problems 3.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLsyst	Crit	cputime
Brock200-1(200)	9	0.1	0.0000e+00	0.0000e+00	0.0000e+00	2.4444	22	1	1.3617e-02
Brock200-2(200)	9	0.1	0.0000e+00	0.0000e+00	0.0000e+00	2.4444	22	1	1.4064e-02
Brock200-3(200)	10	0.1	0.0000e+00	0.0000e+00	0.0000e+00	2.5000	25	1	1.5660e-02
Brock200-4(200)	10	0.1	0.0000e+00	0.0000e+00	0.0000e+00	2.5000	25	1	2.4830e-02
c-fat200-1(200)	8	0.1	4.0000e+00	1.1102e-17	-5.0626e-14	1.8750	15	1	8.2208e-03
c-fat200-2(200)	6	0.1	1.0000e+01	-6.6518e-18	-9.8359e-15	1.5000	9	1	7.4465e-03
c-fat200-5(200)	9	0.1	2.7000e+01	-1.6356e-17	-6.2500e-13	1.4444	13	1	7.6768e-03
Hamming6-2(64)	1	0.1	5.7000e+01	4.1200e-17	-4.9960e-16	1.0000	1	1	3.3304e-04
Hamming6-4(64)	1	0.1	2.2000e+01	1.8431e-18	-3.4694e-17	1.0000	1	1	4.2667e-04
Hamming8-2(256)	1	0.1	2.4700e+02	-1.7889e-17	-2.1372e-15	1.0000	1	1	1.6719e-03
Hamming8-4(256)	1	0.1	1.6300e+02	1.4366e-18	-1.2143e-16	1.0000	1	1	2.1578e-03
Johnson8-2-4(28)	2	0.1	0.0000e+00	0.0000e+00	0.0000e+00	1.5000	3	1	4.2351e-04
Johnson8-4-4(70)	1	0.1	5.3000e+01	6.3441e-18	-1.2490e-16	1.0000	1	1	3.1407e-04
Johnson16-2-4(120)	1	0.1	9.1000e+01	5.8981e-18	-2.6368e-16	1.0000	1	1	5.5862e-04
Johnson32-2-4(496)	1	0.1	4.3500e+02	-3.8612e-18	-7.7022e-16	1.0000	1	1	1.0770e-02
Keller4(171)	6	0.1	0.0000e+00	0.0000e+00	0.0000e+00	1.8333	11	1	6.6781e-03
Mann-a9(45)	8	0.1	8.0000e+00	-6.1679e-18	-2.6701e-14	1.1250	9	1	1.0481e-03
Mann-a27(378)	41	0.1	2.6000e+01	1.7476e-17	-6.1212e-13	1.0244	42	1	3.0620e-01

Table 10.: Performance of ADMM for symmetric EiCP - Test Problems 4.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLsyst	Crit	cputime
RAND(50)	29	70	6.5471e-01	9.0501e-16	-4.6945e-05	1.2759	37	1	7.0511e-03
RAND(100)	47	70	6.6332e-01	-5.3778e-16	-2.8752e-05	1.2128	57	1	1.8022e-02
RAND(250)	27	500	6.5120e-01	-2.5971e-16	-9.2148e-05	1.5926	43	1	4.1763e-02
RAND(500)	30	500	6.7911e-01	5.4191e-16	-1.2419e-05	1.5667	47	1	2.6907e-01
RAND(750)	46	500	6.8664e-01	3.8920e-16	-4.0101e-04	1.2826	59	2	7.3972e-01
RAND(1000)	93	500	6.9763e-01	1.3856e-16	-4.4307e-04	1.3011	121	2	2.4211e+00

Table 11.: Performance of ADMM for symmetric EiCP - Test Problems 5.

Problem	It	ρ	λ	compl	dualfeas	nitBPP	nLsyst	Crit	cputime
RAND(50)	28	70	5.9760e-01	6.8571e-17	-8.5743e-05	1.2500	35	1	7.4880e-03
RAND(100)	64	70	6.2879e-01	3.4937e-16	-4.4386e-05	1.1875	76	1	2.4715e-02
RAND(250)	27	500	6.3822e-01	-9.1566e-18	-8.0979e-05	1.5185	41	1	4.1466e-02
RAND(500)	32	500	6.7176e-01	-1.4312e-16	-1.1228e-04	1.5000	48	2	2.7938e-01
RAND(750)	46	500	6.8151e-01	-1.5786e-16	-5.4690e-04	1.3696	63	2	7.0903e-01
RAND(1000)	99	500	6.9363e-01	1.1627e-16	-3.0355e-04	1.3131	130	2	2.4574e+00

Table 12.: Performance of ADMM for symmetric EiCP - Test Problems 6.

Our experiments with symmetric EiCPs showed that a hybrid method does not seem to be required and lead to our recommendation to use solely this ADMM.

Test Problems 3 were solved by the splitting method A1 in [18]. A report of the performance of this splitting method in that paper clearly indicates that ADMM always requires much less iterations for solving these test problems. By this reason we decided not to solve the remaining test problems by the splitting method and we used another approach for comparison with ADMM. Indeed, it is known [22, 35] that if S is the ordinary simplex given by (5), then a solution $(\bar{x}, \bar{\lambda})$ of the symmetric EiCP can be computed by finding an SP $\bar{x}$ of the following StFQP

$$\min \quad \lambda(x) = \frac{x^\top A x}{x^\top B x} \tag{75a}$$

$$\text{s.t.} \qquad x \in S, \tag{75b}$$

and set $\bar{\lambda} = \lambda(\bar{x})$. So, the complementary eigenvector is the computed SP $\bar{x}$ of StFQP (75) and the complementary eigenvalue is the value of the objective function of this program at the computed SP. Hence, an efficient local optimization solver such as `IPOPT` [38] can be used to solve the symmetric EiCP by computing a stationary point of the program (75). In order to have a better idea of the efficiency of ADMM, we solve all the Test Problems 3 - 6 by `IPOPT`. The corresponding numerical results are showed in Tables 13, 14, 15, and 16, respectively. For Test Problems 3, Algorithm 2 and `IPOPT` compute different complementary eigenvalues and eigenvectors. It is important to add that the complementary eigenvalues computed by IPOPT are always smaller than the ones found by Algorithm 2. This is not surprising, as `IPOPT` is usually able to compute SPs whose objective function values are close to the global minimum value of StFQP. Furthermore, Algorithm 2 computes the same complementary eigenvalues for all Test Problems 4 but two and for all Test Problems 5 and 6. On the other hand, the number of required systems *nLsyst* for Algorithm 2 is usually smaller than the corresponding number of systems *It* that `IPOPT` requires for Test Problems 3 and 4 and bigger for Test Problems 5 and 6. Finally, CPU time is always smaller for Algorithm 2, as the computational effort of each iteration is usually much smaller for this method.

Problem	It	λ	cputime
BCSSTK02(66)	71	6.1532e+00	1.9420e+00
BCSSTK04(132)	29	6.6215e+00	4.0700e-01
BCSSTK05(153)	17	5.5037e+02	4.1300e-01
BCSSTK10(1086)	25	8.5414e+01	4.0200e+00
BCSSTK27(1224)	111	1.9685e+02	2.7830e+01
s1rmq4m1(5489)	31	3.8507e-01	2.2869e+02
s1rmt3m1(5489)	32	3.8507e-01	2.8127e+02
s2rmq4m1(5489)	26	3.9358e-04	2.1173e+02

Table 13.: Performance of `IPOPT` for symmetric EiCP - Test Problems 3.

Problem	It	λ	cputime
Brock200-1(200)	152	0.0000e+00	2.8500e+01
Brock200-2(200)	172	2.0778e-09	1.4450e+00
Brock200-3(200)	125	0.0000e+00	1.1850e+00
Brock200-4(200)	148	0.0000e+00	1.9020e+00
c-fat200-1(200)	261	3.5005e-10	1.7080e+00
c-fat200-2(200)	190	0.0000e+00	1.3810e+00
c-fat200-5(200)	221	0.0000e+00	1.5590e+00
Hamming6-2(64)	4	5.7000e+01	1.9400e-01
Hamming6-4(64)	4	2.2000e+01	1.4900e-01
Hamming8-2(256)	5	2.4700e+02	4.2400e-01
Hamming8-4(256)	5	1.6300e+02	4.2200e-01
Johnson8-2-4(28)	12	0.0000e+00	9.8000e-02
Johnson8-4-4(70)	4	5.3000e+01	1.0300e-01
Johnson16-2-4(120)	5	9.1000e+01	1.2500e-01
Johnson32-2-4(496)	5	4.3500e+02	5.1200e-01
Keller4(171)	49	0.0000e+00	3.9600e-01
Mann-a9(45)	21	8.0000e+00	1.1900e-01
Mann-a27(378)	17	2.6000e+01	6.5700e-01

Table 14.: Performance of `IPOPT` for symmetric EiCP - Test Problems 4.

Problem	It	λ	cputime
RAND(50)	27	6.5471e-01	1.7400e-01
RAND(100)	26	6.6332e-01	1.9900e-01
RAND(250)	12	6.5120e-01	3.8800e-01
RAND(500)	24	6.7911e-01	1.6650e+00
RAND(750)	14	6.8664e-01	2.8020e+00
RAND(1000)	14	6.9763e-01	4.9470e+00

Table 15.: Performance of `IPOPT` for symmetric EiCP - Test Problems 5.

Problem	It	λ	cputime
RAND(50)	26	5.9760e-01	1.5500e-01
RAND(100)	27	6.2879e-01	1.9200e-01
RAND(250)	12	6.3822e-01	3.5700e-01
RAND(500)	25	6.7176e-01	1.7610e+00
RAND(750)	17	6.8151e-01	3.3310e+00
RAND(1000)	14	6.9363e-01	4.8110e+00

Table 16.: Performance of `IPOPT` for symmetric EiCP - Test Problems 6.

As a final conclusion, ADMM seems to be quite efficient for the solution of the symmetric EiCP if an appropriate value of the penalty parameter is chosen. As for the nonsymmetric EiCPs, the use of BPP algorithm for solving the required SCStQPs in each iteration of ADMM and Criterion 1 based on this last method are the two main ingredients for the efficiency of ADMM for dealing with symmetric EiCPs. Furthermore, the penalty parameter ρ should be chosen small and even quite small for some instances, but there are some instances where ADMM requires a larger penalty parameter to terminate.

7. Conclusions

In this paper, we introduce an Alternating Direction Method of Multipliers (ADMM) for finding a solution of the nonsymmetric Eigenvalue Complementarity Problem (EiCP). A partial convergence analysis of the algorithm is presented and shows that the limit point of the sequence of iterates is a solution of the nonsymmetric EiCP provided the sequence converges. A simpler form of ADMM for finding a solution of the symmetric EiCP is also discussed. It is shown that for this case an accumulation point x^* the sequence of iterates $\{x^k\}$ gives a solution $(x^*, \lambda(x^*))$ of the symmmetric EiCP with $\lambda(x)$ given by (2), provided two conditions are satisfied. Moreover, an extension of ADMM for computing a Stationary Point (SP) of a Standard Quadratic Program (StQP) is introduced together with a similar convergence result.

A Block Principal Pivoting (BPP) and a Stopping Criterion based on this algorithm are the two most important ingredients for the good performance of ADMM to solve small and large dimensional EiCPs. Computational experiments reported in this paper indicate that ADMM can be slow and may face difficulties for terminating for solving some nonsymmetric EiCPs. However, the hybridization with a Semi-smooth Newton (SN) method seems to work well in practice. On the other hand, ADMM seems to solve efficiently the symmetric EiCP.

Computational experiments also indicate that the penalty parameter has an important effect on the performance of ADMM. An automatic procedure for choosing this penalty parameter is an important issue for future research. Finally, the use of ADMM for computing efficiently an SP of an StQP in practice should also be investigated.

References

[1] Abdi, F., Shakeri, F., “A new descent method for symmetric nonmonotone variational inequalites with applications to the eigenvalue complementarity problem.” Journal of Optimization Theory and Applications 173, 923-940 (2017).

[2] Adly, S., Rammal, H., “A new method for solving second-order cone eigenvalue complementarity problem.” Journal of Optimization Theory and Applications 165, 563-585 (2015).

[3] Adly, S., Seeger, A., “A nonsmooth algorithm for cone constrained eigenvalue problems.” Computational Optimization and Applications 49, 299-318 (2011).

[4] Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning 3, 1-122 (2011).

[5] Bot, R. I. and Nguyen, D.-K., “The proximal alternating direction method of multiplies in the nonconvex setting: convergence analysis and rates,” arXiv:1801.01994 (2018).

[6] Brás, C., Fukushima, M., Júdice, J. J., Rosa, S., “Variational inequality formulation for the asymmetric eigenvalue complementarity problem and its solution by means of a gap function.” Pacific Journal of Optimization 8, 197-215 (2012).

[7] Brás, C., Iusem, A. N., Júdice, J. J., “On the quadratic eigenvalue complementarity problem.” Journal of Global Optimization 66, 153-171 (2016).

[8] Brás, C., Fischer, A., Júdice, J. J., Schönefeld, K., Seifert, S., “A block active set algorithm with spectral choice line search for the symmetric eigenvalue complementarity problem.” Applied Mathematics and Computation 294, 36-48 (2017).

[9] Chen, Z. M., Qi, L. Q., “A semismooth Newton method for tensor eigenvalue complemetarity problem.” Computational Optimizarion and Applications 165, 583-585 (2015).

[10] DIMACS: Second DIMACS Challenge. Test instances available at `http://dimacs.rutgers.edu/challenges`.

[11] Eckstein, J., “Augmented Lagrangian and Alternating Direction Methods for Convex Optimization: A Tutorial and Some Illustrative Computational Results,” RUTCOR Research Report RRR 32-2012, Rutgers University (2012).

[12] Facchinei, F. and Pang, J. S., *Finite-Dimensional Variational Inequalities and Complementarity Problems.* Springer, Berlin (2003).

[13] Fernandes, L. M., Júdice, J. J, Sherali, H. D., Fukushima, M.,“On the computation of all eigenvalues for the eigenvalue complementarity problem.” Journal of Global Optimization, 59, 307-326 (2014).

[14] Fernandes, R,, Júdice, J. J, Trevisan, V., “Complementary eigenvalues of graphs.” Linear Algebra and Its Applications 527, 216-231 (2017).

[15] Fun, J., Nie, J., Zhou, A.,“Tensor eigenvalue complementarity problems.” Mathematical Programming Series A 170, 507-539 (2018).

[16] G. Golub and C. Van Loan, *Matrix Computations.* The John Hopkins University Press, Baltimore (1996).

[17] Holubova, G., Necedal, P., “A note on a relation between the Fucik spectrum and Pareto eigenvalues.” Journal of Mathematical Analysis and Applications 427, 618-628 (2015).

[18] Iusem, A. N., Júdice, J. J., Sessa, V., and Sarabando, P., “Splitting methods for the Eigenvalue Complementarity Problem,” Optimization Methods and Software 34, 1184-1212 (2019).

[19] Iusem, A. N., Júdice, J. J., Sessa, V., Sherali, H., “The second-order cone quadratic eigenvalue complementarity problem.” Pacific Journal of Optimization 13, 475-500 (2017).

[20] Júdice, J. J, Pires, F., “A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems.” Computers and Operations Research 21, 587-596 (1994).

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

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

[23] LeThi, H. A., Moeini, M., Pham, D. T., Júdice, J. J., “A DC programming apporach for solving the symmetric eigenvalue complementarity problem.” Computational Optimization and Applications 51, 1097-1117 (2012).

[24] Li, G. and Pong, T.K. “Global convergence of splitting methods for nonconvex composite optimization,” SIAM Journal on Optimization 25, 2434-2460 (2015).

[25] Murty, K., *Linear Complementarity, Linear and Nonlinear Programming.* Heldermann Verlag, Berlin (1988).

[26] Niu, Y. S., LeThi, H. A., Pham, D. T., Júdice, J. J., "Efficient DC programming approaches for solving the asymmetric eigenvalue complementarity problem." Optimization Methods and Software 28, 812-829 (2013).

[27] Nocedal, J. and Wright, S., *Numerical Optimization.* Springer, New York (2006).

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

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

[30] Pinto da Costa, A., Seeger, A., Simões, F. M. F., "Complementarity eigenvalue problems for nolinear matrix pencils." Applied Mathematics & Computation 312, 134-148 (2017).

[31] Queiroz, M., Júdice, J., and Humes, C., "The symmetric eigenvalue complementarity problem." Mathematics of Computation 73, 1849-1863 (2003).

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

[33] Seeger, A., "Complementarity eigenvalue analysis of connected graphs." Linear Algebra and Its Applications 543, 205-225 (2018).

[34] Seeger, A., Torki, M., "On eigenvalues induced by a cone constraint." Linear Algebra and Its Applications 372, 181-206 (2003).

[35] Seeger. A., Torki, M.,"Local minima of quadratic forms on convex cones." Journal of Global Optimization 44, 1-28 (2009).

[36] Seeger, A., Vicente-Perez, J. "On cardinality of Pareto spectra." The Electronic Linear Algebra Journal 22, 758-766 (2011).

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

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

[39] Yu, G., Song Y., Xu, Y., "Spectral projected gradient methods for generalized tensor eigenvalue complementarity problems." Numerical Algorithms 80, 1181-1201 (2019).

[40] Zhang, L., Shen, C., Yang, M., Júdice, J. J., "A Lanczos method for large-scale extreme Lorentz eigenvalue problems." SIAM Journal on Matrix Analysis and Applications 39, 611-631 (2018).

[41] Zhou, Y. H., Gowda, M. S., "On the finiteness of the cone spectrum of certain linear transformations on Euclidean Jordan algebras." Linear Algebra and Its Applications 431, 772-782 (2009).