Numer Algor (2008) 47:391–407
DOI 10.1007/s11075-008-9194-7

ORIGINAL PAPER

On the solution of the symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm

Joaquim J. Júdice · Marcos Raydan · Silvério S. Rosa · Sandra A. Santos

Received: 22 October 2007 / Accepted: 29 February 2008 /
Published online: 29 March 2008

Abstract This paper is devoted to the eigenvalue complementarity problem (EiCP) with symmetric real matrices. This problem is equivalent to finding a stationary point of a differentiable optimization program involving the Rayleigh quotient on a simplex (Queiroz et al., Math. Comput. 73, 1849–1863, 2004). We discuss a logarithmic function and a quadratic programming formulation to find a complementarity eigenvalue by computing a stationary point of an appropriate merit function on a special convex set. A variant of the spectral projected gradient algorithm with a specially designed line search is introduced to solve the EiCP. Computational experience shows that the application of this algorithm to the logarithmic function formulation is a quite efficient way to find a solution to the symmetric EiCP.

Keywords Complementarity · Projected gradient algorithms · Eigenvalue problems

J. J. Júdice
Dpto. de Matemática, Universidade de Coimbra, Coimbra, Portugal
e-mail: joaquim.judice@co.it.pt

M. Raydan (✉)
Dpto. de Computación, Universidad Central de Venezuela,
Ap. 47002, Caracas 1041-A, Venezuela
e-mail: mraydan@kuaimare.ciens.ucv.ve

S. S. Rosa
Dpto. de Matemática, Universidade da Beira Interior, Covilhã, Portugal
e-mail: rosa@mat.ubi.pt

S. A. Santos
Dpto. de Matemática Aplicada, IMECC-UNICAMP,
CP 6065, 13081-970 Campinas SP, Brazil
e-mail: sandra@ime.unicamp.br

1 Introduction

Given the matrix $A \in \mathbb{R}^{n\times n}$ and the positive definite (PD) matrix $B \in \mathbb{R}^{n\times n}$, the eigenvalue complementarity problem (EiCP) is a problem of the form

$$\text{Find } \lambda > 0 \text{ and } x \in \mathbb{R}^n \setminus \{0\} \text{ such that } \begin{cases} w = (\lambda B - A)x, \\ w \geqslant 0,\ x \geqslant 0, \\ x^T w = 0. \end{cases} \tag{1.1}$$

The EiCP is a particular case of the mixed eigenvalue complementarity problem (MEiCP$_J$) that consists of finding a scalar $\lambda > 0$ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$\begin{cases} w = (\lambda B - A)x, \\ w_J \geqslant 0,\ x_J \geqslant 0, \\ w_J^T x_J = 0, \\ w_{\bar{J}} = 0, \end{cases}$$

where $x_J \equiv (x_j,\ j \in J)$, $w_J \equiv (w_j,\ j \in J)$, $J \subseteq \{1, \ldots, n\}$ and $\bar{J} = \{1, \ldots n\} \setminus J$. Note that the EiCP is obtained when $J = \{1, \ldots, n\}$. The MEiCP$_J$ is a generalization of the EiCP, that appears more frequently in practical problems of engineering and physics where the computation of eigenvalues is required. Problems involving the resonance frequency of structures and stability of dynamical systems are among these applications and have been discussed in [9]. Extensions of these problems to more general cones have been discussed in [24–26, 28]. We are interested in the Symmetric EiCP, in which the matrices A and B are both symmetric (i.e., when B is SPD). As is traditional in complementarity problems, the most important conclusions for the EiCP also hold for the MEiCP$_J$.

Note that if λ is unrestricted and $w = 0$ ($J = \emptyset$), then the variables x_i ($i = 1, \ldots, n$) are free and the solution of the MEiCP$_J$ corresponds to a solution of the Generalized Eigenvalue Problem [15]. For any solution (λ, x) of EiCP (or MEiCP$_J$), the value of λ is called *Complementary Eigenvalue* of the matrices (A, B) and x is the corresponding *Complementary Eigenvector.*

For each solution (λ, x) of MEiCP$_J$, there exists a set of indices I satisfying $\bar{J} \subseteq I \subseteq \{1, \ldots, n\}$, such that λ is a positive eigenvalue of (A_{II}, B_{II}) and x_I is the corresponding eigenvector satisfying $x_{J\cap I} \geqslant 0$ [24], where C_{II} represents the principal submatrix of order I of the matrix C and x_I is the subvector associated with the index set I. For the EiCP, this theorem means that given a solution (λ, x), λ is a positive eigenvalue of (A_{II}, B_{II}) and x_I is the corresponding non-zero eigenvector. An immediate corollary of this result is that the number of solutions of the EiCP (and MEiCP$_J$) is finite [22, 24].

When at least one of the matrices A or B is asymmetric, the EiCP was studied in [17], where a branch-and-bound method for the solution of this problem was introduced. The symmetric EiCP, as defined by (1.1), was discussed in [22], where it was shown that the EiCP can be reduced to the problem of finding a stationary point of the Rayleigh function on the simplex.

In this paper we start by recalling the optimization formulation that uses the Rayleigh quotient function. We also consider a logarithmic function applied to the Rayleigh quotient. A quadratic formulation equivalent to (1.1) is also introduced. The resulting problems are nonlinear programs that can be solved by an interior-point method such as `LOQO` [27], and also by a general purpose optimization solver as `MINOS` [20]. For the Rayleigh quotient and the logarithmic function, the EiCP is reduced to nonlinear programs on a simplex. We discuss the solution of these two optimization problems by a variant of the spectral projected gradient (SPG) [6] algorithm combined with a specially designed line search, fully described in Section 3. The projection, required at each iteration of this process, is the unique optimal solution of a strictly convex quadratic program solved by a strongly polynomial block pivotal principal pivoting algorithm [16]. Computational experience with a set of small and large EiCPs shows that the SPG algorithm is quite efficient for finding a complementary eigenvalue and compares favorably with the commercial codes `LOQO` and `MINOS` in these instances. Furthermore the logarithmic function formulation seems to lead in general into a better performance for the SPG algorithm.

The paper is organized as follows. In Section 2 the three formulations are presented. The SPG method is introduced in Section 3 along with all the techniques incorporated in the algorithm for the computation of the search direction and step length. Numerical experiments and some conclusions are presented in the last section of this paper.

2 Formulations

Since the set of complementary eigenvectors associated to a certain eigenvalue is a cone, there is no loss of generality if we consider only the solutions satisfying $\|x\| = p$, where $p > 0$ and $\| \ \|$ is any vector norm. This constraint ensures that x is a non-zero vector. Since $x \geqslant 0$ in the definition of the EiCP, this constraint can be replaced by the linear constraint $\|x\|_1 = e^T x = p$, where e is a vector of ones. So (1.1) is equivalent to finding $\lambda > 0$ such that

$$\begin{aligned} w &= (\lambda B - A)x, \\ w &\geqslant 0, \ x \geqslant 0, \\ x^T w &= 0, \\ e^T x &= p. \end{aligned} \tag{2.1}$$

Considering a suitable continuously differentiable merit function $\phi(x)$ [22], it is possible to reduce the EiCP to the following nonlinear program

$$\begin{aligned} \text{Minimize} \quad & \phi(x) \\ \text{subject to} \quad & e^T x = p, \\ & x \geqslant 0. \end{aligned} \tag{2.2}$$

The Karush–Kuhn–Tucker conditions that define a stationary point for this problem constitute the complementarity problem

$$
\begin{aligned}
\nabla\phi(x) + \alpha e &= w, \\
e^T x &= p, \\
x^T w &= 0, \\
w \geqslant 0,\ x \geqslant 0,\ &\alpha \in \mathbb{R},
\end{aligned} \tag{2.3}
$$

where α is the Lagrange multiplier associated to the constraint $e^T x = p$.

2.1 Rayleigh quotient formulation

The first objective function is the generalized Rayleigh quotient that was used in [22]. It is included in this work for completeness.

The complementarity condition $w^T x = 0$ in (2.1) may be substituted by $x^T(\lambda Bx - Ax) = 0$ and, since B is SPD, this equation is equivalent to

$$
\lambda(x) = \frac{x^T Ax}{x^T Bx}.
$$

This is the generalized Rayleigh quotient.

As discussed in [22], if

$$
\phi(x) = -\frac{x^T Ax}{x^T Bx},
$$

then a stationary point of (2.2) gives a solution to the EiCP. The gradient and Hessian for this function are respectively

$$
\nabla\phi(x) = \frac{2}{(x^T Bx)^2}\left(\left(x^T Ax\right) Bx - \left(x^T Bx\right) Ax\right), \tag{2.4}
$$

and

$$
\begin{aligned}
\nabla^2\phi(x) = &\frac{2\left[\left(x^T Ax\right) B - \left(x^T Bx\right) A\right] + 4\left[(Ax)(Bx)^T + (Bx)(Ax)^T\right]}{\left(x^T Bx\right)^2} \\
&- \frac{8\left(x^T Ax\right)(Bx)(Bx)^T}{\left(x^T Bx\right)^3}.
\end{aligned} \tag{2.5}
$$

2.2 Logarithmic formulation

Inspired by the work of Auchmuty [1], Mongeau and Torki [19] and the behavior of the generalized Rayleigh quotient, we introduce the following merit function

$$
L_{AB}(x) = \ln\left(x^T Bx\right) - \ln\left(x^T Ax\right),
$$

whose gradient and Hessian are respectively

$$
\nabla L_{AB}(x) = \frac{2Bx}{x^T Bx} - \frac{2Ax}{x^T Ax}, \tag{2.6}
$$

and

$$\nabla^2 L_{AB}(x) = \frac{2B}{x^T Bx} - \frac{4(Bx)(Bx)^T}{(x^T Bx)^2} - \frac{2A}{x^T Ax} + \frac{4(Ax)(Ax)^T}{(x^T Ax)^2}. \tag{2.7}$$

Note that this function can only be used if $x^T Ax > 0$ for any $x \neq 0$, that is, if A is strictly copositive [10]. Moreover, $x^T Bx > 0$ for any $x \neq 0$, since B is SPD.

Theorem 2.1 *If A is strictly copositive, then any stationary point $\bar{x}$ of $L_{AB}(x)$ in the convex set $K = \{x \in \mathbb{R}^n : e^T x = p, x \geqslant 0\}$ leads to the solution $(\bar{x}, \bar{\lambda})$ of the EiCP, where $\bar{\lambda} = (\bar{x}^T A\bar{x})/(\bar{x}^T B\bar{x})$.*

Proof Computing the inner product of $x \in \mathbb{R}^n$ and the gradient vector, we obtain

$$x^T \nabla L_{AB}(x) = 0. \tag{2.8}$$

By assuming that $(\bar{x}, \bar{w}, \bar{\alpha}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$ satisfy the conditions (2.3) with $\phi(x) = L_{AB}(x)$, then

$$\bar{x}^T \nabla L_{AB}(\bar{x}) + \bar{\alpha}\left(\bar{x}^T e\right) = \bar{x}^T \bar{w} = 0.$$

By (2.8) and since $\bar{x}^T e = p > 0$ it follows that $\bar{\alpha} = 0$. Now by (2.3) we have

$$\nabla L_{AB}(\bar{x}) = \bar{w} \geqslant 0. \tag{2.9}$$

Since A is strictly copositive and B is SPD, the expressions (2.6) and (2.9) imply that $(\bar{\lambda}B - A)\bar{x} \geqslant 0$ with $\bar{\lambda} = (\bar{x}^T A\bar{x})/(\bar{x}^T B\bar{x})$ and $\bar{x}^T(\bar{\lambda}B - A)\bar{x} = 0$. Hence $(\bar{x}, \bar{\lambda})$ is a solution of the EiCP. □

2.3 Quadratic formulation

An equivalent way to formulate the EiCP (1.1) is through the quadratic formulation:

$$\begin{aligned} &\text{Maximize} \quad x^T Ax \\ &\text{subject to} \quad x^T Bx \leqslant 1, \\ &\qquad\qquad\qquad x \geqslant 0, \end{aligned} \tag{2.10}$$

where the matrix A is symmetric and B is SPD.

Theorem 2.2 *If A is strictly copositive and $\bar{x} \neq 0$ is a stationary point of (2.10), then the pair $\bar{x}$, $\bar{\lambda} = \bar{x}^T A\bar{x}$ is a solution of EiCP.*

Proof Since the constraint set is convex, we start by showing that Slater's constraint qualification [3] holds, that is, there exists an $x > 0$ such that $x^T Bx < 1$.

Since B is an SPD matrix, then $B \in S$ [10] and there exists an $\bar{x} > 0$ such that $B\bar{x} > 0$. Furthermore, $\bar{x}^T B\bar{x} > 0$ and there are three possible cases:

a) $\bar{x}^T B\bar{x} < 1$ and the Slater's constraint qualification holds;
b) $\bar{x}^T B\bar{x} = 1$;
c) $\bar{x}^T B\bar{x} = \beta > 1$.

The third case reduces to the second one, as $\tilde{x} = \bar{x}/\sqrt{\beta}$ satisfies $\tilde{x} > 0,\ B\tilde{x} > 0$ and $\tilde{x}^T B\tilde{x} = 1$.

Consider now that there exists $\bar{x} > 0$ such that $B\bar{x} > 0$ and $\bar{x}^T B\bar{x} = 1$. We prove that for any positive real number θ such that

$$\theta < \min\left\{\bar{x}_1, \frac{2(B\bar{x})_1}{b_{11}}\right\}$$

then $x = \bar{x} - \theta e^1$ satisfies $x > 0$ and $x^T Bx < 1$ (e^1 is the first vector of the canonical basis). In fact $x > 0$ by construction. Furthermore

$$\begin{aligned} x^T Bx &= (\bar{x} - \theta e^1)^T B(\bar{x} - \theta e^1) \\ &= \bar{x}^T B\bar{x} - 2\theta(B\bar{x})_1 + \theta^2 b_{11}, \end{aligned}$$

and $x^T Bx < 1$ if and only if

$$\left(\bar{x}^T B\bar{x} - 1\right) + \theta[\theta b_{11} - 2(B\bar{x})_1] < 0,$$

that is, if and only if, $\theta < 2(B\bar{x})_1/b_{11}$. This shows that Slater's constraint qualification is true and [3] any optimal solution $\bar{x}$ of (2.10) satisfies the Karush–Kuhn–Tucker (KKT) conditions

$$\begin{aligned} w &= (\lambda B - A)x, \\ x_i w_i &= 0,\ i = 1, \ldots, n \\ x &\geqslant 0, w \geqslant 0, \lambda \geqslant 0, \\ \nu &= 1 - x^T Bx, \\ \nu\lambda &= 0, \\ \nu &\geqslant 0. \end{aligned}$$

Since $0 \neq \bar{x} \geqslant 0$ and B is an SPD matrix, then $\bar{x}^T B\bar{x} > 0$. If $\bar{x}^T B\bar{x} < 1$, then $\lambda = 0$ and $\bar{w} = -A\bar{x}$. Therefore

$$\bar{x}^T \bar{w} = 0 = -\bar{x}^T A\bar{x},$$

which is impossible, since A is strictly copositive. Hence $\bar{x}^T B\bar{x} = 1$ and $\lambda = \bar{x}^T A\bar{x} > 0$. This completes the proof. □

3 Spectral projected gradient algorithm

In this section the spectral projected gradient (SPG) method is applied to the two formulations with linear constraints of the previous section. This method can be viewed as a variant of the classical projected gradient method.

Projected gradient (PG) methods provide an interesting option for solving large-scale convex constrained problems. They are simple and easy to code, and avoid the need for matrix factorizations. Practical monotone backtracking line search versions have been introduced to the choice of step length (see e.g., [4]). However, these early PG methods are frequently inefficient since their performance resembles the optimal gradient method (also known as the steepest descent method), which is usually very slow. Nevertheless, the effectiveness of PG methods can be greatly improved by incorporating recently developed choices of step length and globalization strategies.

There have been many different variations of the early PG methods. They all have the common property of maintaining feasibility of the iterates by frequently projecting trial steps on the feasible convex set. In particular, Birgin et al. [5, 6] combine the projected gradient method with recently developed ingredients in unconstrained optimization to propose an effective scheme that is known as the spectral projected gradient (SPG) method. One of the interesting features of the SPG method is the spectral choice of step length along the gradient direction, originally proposed by Barzilai and Borwein [2] for unconstrained optimization. In [2], R-superlinear convergence was established for the minimization of two-dimensional strictly convex quadratics. Recently, though, Dai and Fletcher [11] established that the method is also asymptotically R-superlinearly convergent in the three-dimensional case, but not when the dimension is greater than or equal to four. Dai and Liao [12] refined the global analysis in Raydan [23] for quadratics and proved that the convergence rate is R-linear in general. Numerical experiments have shown that the spectral gradient method for unconstrained optimization ([13]) or the SPG method for convex constrained optimization ([5]) are much faster than the steepest descent method or the classical PG methods, respectively.

In the setting of Birgin et al. [5, 6], the SPG algorithm starts with $x_0 \in \Omega$, and moves at every iteration k along the internal projected gradient direction

$$d_k = P(x_k - \eta_k \nabla\phi(x_k)) - x_k,$$

with a parameter $\eta_k > 0$. In particular, in the SPG method, the spectral choice is used, which is given by

$$\eta_k = \frac{\langle s_{k-1}, s_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle},$$

$s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = \nabla\phi(x_k) - \nabla\phi(x_{k-1})$ whenever the denominator is positive. Furthermore, $P(w)$ is the projection of $w \in \mathbb{R}^n$ onto Ω, where for the optimization formulations under study

$$\Omega = \left\{x \in \mathbb{R}^n : x \geqslant 0, e^T x = p\right\}. \tag{3.1}$$

In the case that the first trial point, $x_k + d_k$, is rejected the next ones are computed along the same direction, i.e., $x_+ = x_k + \delta d_k$, using a line search to choose $0 < \delta \leq 1$, to be described later, such that global convergence towards a stationary point of ϕ is guaranteed.

We now present the algorithm used in this paper. It starts with $x_0 \in \Omega$, a sufficient decrease parameter $\zeta \in (0, 1)$, and a small stopping tolerance $\epsilon > 0$. Initially, $\eta_0 > 0$ is arbitrary. Given $x_k \in \Omega$ and $\eta_k > 0$, we describe next an iteration of the SPG algorithm.

Spectral projected gradient algorithm

Step 1 Compute $z_k = P(x_k - \eta_k \nabla\phi(x_k))$ and the direction $d_k \in \mathbb{R}^n$ by

$$d_k = z_k - x_k.$$

Step 2 If $\|d_k\| < \epsilon$ then stop: x_k is a stationary point of ϕ in Ω.

Step 3 If $\phi(z_k) \leqslant \phi(x_k) - \zeta d_k^T \nabla\phi(x_k)$ then $\delta_k = 1$.
Else determine the step length $\delta_k \in]0; 1]$ by exact line search.

Step 4 Update the solution

$$x_{k+1} \leftarrow x_k + \delta_k d_k.$$

In the implementation, the value of ϵ depends on the optimization problem. This value must guarantee that the algorithm ends after a finite number of iterations and the solution is accurate.

At each iteration we have to compute the objective function, its gradient and the projection z_k. The gradient is given by (2.4) or (2.6), depending on the merit function to be used. We now discuss how to obtain the initial solution, the parameter η_k, the step length δ_k, and the projection.

3.1 The initial guess

As described in [22], the initial solution x_0 can be chosen by one of several processes. In particular if A has at least one diagonal element $a_{ii} > 0$ then the initial solution can be chosen as

$$x_0 = p\, e^i,$$

where e^i is the vector i of the canonical basis. Another possible choice is

$$x_0 = \frac{p}{n} e, \tag{3.2}$$

as long as $(x_0)^T A x_0 > 0$. Therefore this initial point can be used if A is strictly copositive.

3.2 The parameter η_k

The parameter η_k can be fixed or changed at each iteration. A first choice is simply setting $\eta_k = 10^{-1}$ or any other small positive value.

A second choice demands some computational effort and is based on [6]. When calculating the first projection we begin with

$$\eta_0 = \min(\eta_{\max}, \max(\eta_{\min}, 1/\| P(x_0 - \nabla\phi(x_0)) - x_0\|_\infty)),$$

where $\eta_{\min}$ is a quite small positive number and $\eta_{\max} = \eta_{\min}^{-1}$. The subsequent values are obtained by the following procedure:

$\texttt{Compute}\ s_k = x_{k+1} - x_k,\ y_k = \nabla\phi(x_{k+1}) - \nabla\phi(x_k)\ \texttt{and}\ \beta_k = \langle s_k, y_k\rangle.$

$\texttt{If}\ \beta_k \leqslant 0\ \texttt{then}\ \eta_{k+1} = \eta_{\max},$

$\texttt{else compute}\ \eta_{k+1} = \min[\eta_{\max}, \max(\eta_{\min}, \langle s_k, s_k\rangle/\beta_k)]$

This process is well-defined [6]. In practice, we accept $\eta_{k+1} = \eta_{\max}$ provided $\beta_k \leqslant \epsilon$ with ϵ a positive tolerance. Furthermore, we set $\eta_{\min} = \epsilon_M$, where ϵ_M is the machine precision.

3.3 Projecting onto Ω

Next, we explain the computation of the projection $z = P(x_k - \eta_k \nabla\phi(x_k))$ onto Ω, given by (3.1), that is required in every iteration of the algorithm.

I. Find $u = x_k - \eta_k \nabla\phi(x_k)$.
II. The vector z is the unique optimal solution of the strictly convex quadratic problem

$$\begin{aligned} &\underset{z\in\mathbb{R}^n}{\text{Minimize}}\ \frac{1}{2}\|u - z\|_2^2 \\ &\text{subject to} \\ &\qquad e^T z = p, \\ &\qquad z_i \geqslant 0,\ i = 1, \ldots, n. \end{aligned}$$

Since

$$\|u - z\|_2^2 = (u - z)^T (u - z) = u^T u - 2u^T z + z^T z,$$

and $\frac{1}{2}u^T u$ is constant, then this program reduces to

$$\begin{aligned} &\underset{z\in\mathbb{R}^n}{\text{Minimize}}\ q^T z + \frac{1}{2} z^T z \\ &\text{subject to} \\ &\qquad e^T z = p, \\ &\qquad z_i \geqslant 0, \quad i = 1, \ldots, n. \end{aligned} \tag{3.3}$$

where $q = -u$.

Several methods can be used to solve this kind of quadratic programs. Among those methods, the block pivotal principal pivoting algorithm presented in [16] is chosen because it is strongly polynomial and very efficient. The steps of this method are presented below.

Block pivotal principal pivoting algorithm

Step 0 Let $F = \{1, 2, \ldots, n\}$.

Step 1 Compute $\varphi = -\dfrac{p + \sum_{i\in F} q_i}{|F|}$.

Step 2 Let $H=\{i\in F: q_i+\varphi>0\}$.
If $H=\varnothing$ stop and

$$z=(z_i)_{i=1,\ldots,n},\ \text{where } z_i=\begin{cases}0 & \text{if } i\notin F\\ -(q_i+\varphi) & \text{if } i\in F\end{cases}$$

is the optimal solution of the quadratic program (3.3).
Otherwise, set $F=F-H$ and return to **Step 1**.

Note that, in practice, the set H is determined by

$$H=\{i\in F: q_i+\varphi>\epsilon\},$$

where $\epsilon=\sqrt{\epsilon_M}$.

3.4 The step length δ

If the first trial point does not satisfy the condition in Step 3 of the SPG algorithm, then the value of the step length δ is obtained by exact line search, i.e., it is the solution of the unconstrained program:

$$\begin{array}{l}\text{Minimize } \phi(x+\delta d)=\varphi(\delta),\\ 0\leqslant\delta\leqslant 1\end{array}\tag{3.4}$$

which depends on ϕ. Next, we explain how δ can be computed for the Rayleigh quotient and logarithmic functions.

Rayleigh quotient function In this case,

$$\varphi(\delta)=-\frac{(x+\delta d)^TA(x+\delta d)}{(x+\delta d)^TB(x+\delta d)}.$$

Here we ignored any reference (subindex) to the iteration number.

Theorem 3.1 *Any solution δ of the equation $\varphi'(\delta)=0$ associated with the Rayleigh quotient merit function is a root of the equation of degree two:*

$$a_1+\delta a_2+\delta^2a_3=0,$$

where

$$\begin{aligned}a_1&=\left(d^TAx\right)\left(x^TBx\right)-\left(d^TBx\right)\left(x^TAx\right),\\ a_2&=\left(d^TAd\right)\left(x^TBx\right)-\left(d^TBd\right)\left(x^TAx\right),\\ a_3&=\left(d^TAd\right)\left(x^TBd\right)-\left(d^TBd\right)\left(x^TAd\right).\end{aligned}\tag{3.5}$$

Proof The stationary point of problem (3.4) satisfies

$$\varphi'(\delta)=\frac{2d^T B(x+\delta d)(x+\delta d)^T A(x+\delta d)-2d^T A(x+\delta d)(x+\delta d)^T B(x+\delta d)}{[(x+\delta d)B(x+\delta d)]^2}=0$$

$$\Leftrightarrow \frac{2d^T A(x+\delta d)(x+\delta d)^T B(x+\delta d)}{[(x+\delta d)B(x+\delta d)]^2}=\frac{2d^T B(x+\delta d)(x+\delta d)^T A(x+\delta d)}{[(x+\delta d)B(x+\delta d)]^2}. \tag{3.6}$$

Since the matrix B is SPD and $x \neq 0$, then (3.6) is equivalent to:

$$d^T A(x+\delta d)(x+\delta d)^T B(x+\delta d) = d^T B(x+\delta d)(x+\delta d)^T A(x+\delta d). \tag{3.7}$$

Simplifying the left side of (3.7), we get

$$\begin{aligned} & d^T A(x+\delta d)(x+\delta d)^T B(x+\delta d) = \\ & = \left[d^T Ax+\delta d^T Ad\right]\left[x^T Bx+2\delta x^T Bd+\delta^2 d^T Bd\right] \\ & = \left(d^T Ax\right)\left(x^T Bx\right)+2\delta\left(x^T Bd\right)\left(d^T Ax\right)+\delta^2\left(d^T Bd\right)\left(d^T Ax\right)+ \\ & \quad +\delta\left(d^T Ad\right)\left(x^T Bx\right)+2\delta^2\left(d^T Ad\right)\left(x^T Bd\right)+\delta^3\left(d^T Ad\right)\left(d^T Bd\right). \end{aligned} \tag{3.8}$$

Furthermore, the right side of (3.7) leads to

$$\begin{aligned} & d^T B(x+\delta d)(x+\delta d)^T A(x+\delta d) = \\ & = \left[d^T Bx+\delta d^T Bd\right]\left[x^T Ax+2\delta x^T Ad+\delta^2 d^T Ad\right] \\ & = \left(d^T Bx\right)\left(x^T Ax\right)+2\delta\left(x^T Ad\right)\left(d^T Bx\right)+\delta^2\left(d^T Ad\right)\left(d^T Bx\right)+ \\ & \quad +\delta\left(d^T Bd\right)\left(x^T Ax\right)+2\delta^2\left(d^T Bd\right)\left(x^T Ad\right)+\delta^3\left(d^T Bd\right)\left(d^T Ad\right). \end{aligned} \tag{3.9}$$

From (3.8), (3.9) and recalling that A and B are symmetric matrices, we obtain

$$a_1+\delta a_2+\delta^2 a_3=0, \tag{3.10}$$

for a_1, a_2 and a_3 as in (3.5). □

Now let s_1 and s_2 be the solutions of equation (3.10). Since $0 \leqslant \delta \leqslant 1$, then the step length δ in the SPG algorithm is computed as follows:

1) $s_1, s_2 \notin [0,1] \Rightarrow \delta = 1$
2) There exists only one $s_i \in [0,1]$, $i \in \{1,2\} \Rightarrow \delta = s_i$
3) $s_1, s_2 \in [0,1]$.
   Then, $\delta = \begin{cases} s_1, & \text{if } \varphi(s_1) \leqslant \varphi(s_2) \\ s_2, & \text{otherwise.} \end{cases}$

Logarithmic function In this case,

$$\varphi(\delta)=\phi(x+\delta d)=log((x+\delta d)^T B(x+\delta d))-log((x+\delta d)^T A(x+\delta d)).$$

and the following result holds.

Theorem 3.2 *Any solution δ of the equation $\varphi'(\alpha)=0$ satisfies the polynomial equation of second degree* (3.10).

The proof is identical to the previous case and the step length δ is obtained following the same steps.

4 Numerical results

The computational experience presented in this section was done on a personal computer with 3.0 GHz Pentium IV processor and 2 GBytes of RAM memory, running Linux 2.6.22. The `MINOS` code of `GAMS` [7] and `LOQO` code of `AMPL` [14] collections were used to solve the three nonlinear formulations. Moreover, the SPG algorithm was used for the first two formulations and implemented in FORTRAN 90 [8], using the Intel compiler, version 10.0. Running times presented in this section are always given in CPU seconds. The times reported for the SPG algorithm were measured using the `system_clock` intrinsic subroutine.

For our initial test problems, the matrix B is, by default, the identity matrix and the matrix $A \in \mathbb{R}^{n\times n}$ is SPD and sparse (`pentadiagonal` [21, page 380]) or dense (`Fathy` [21, page 311]). It is interesting to note that matrices A of `Fathy` class are all positive. This means that each one of these matrices has exactly one positive complementarity eigenvalue, which is its Perron root [24]. On the other hand this uniqueness property no longer holds for the matrices of the `pentadiagonal` class. The parameter p in the constraint $e^T x = p$ has the fixed value $p = 1$. In our experiments, we fix the parameters $\epsilon = 10^{-6}$ and $\zeta = 10^{-4}$.

The test problems are scaled according to the procedure described in [17, Section 5]. The scaling is important because the matrices that we are using are badly conditioned, and without this tool some of the problems cannot be solved.

Table 1 contains the results of the SPG algorithm for solving symmetric EiCPs with the initial solution (3.2), the Rayleigh quotient and the logarithmic objective functions. In our tests with these matrices the algorithm had better results with this initial solution. In the referred Table, '`Rayleigh`' means 'Rayleigh quotient function', '`Logarithmic`' means 'logarithmic function', '`λ`' is the complementarity eigenvalue found for the EiCP, '`T`' is the total CPU time performed by the method and '`It`' is the number of iterations needed to solve each problem. This notation is also used in the remaining tables.

Table 1 shows that usually the logarithmic function makes the SPG method slightly more efficient when A is the `pentadiagonal` matrix. We also observe that the algorithm has an identical performance for both objective functions for matrices of Fathy class. For the pentadiagonal matrices, the number of iterations increases much with the dimension of the EiCP. However, it is noticed that the SPG method can perform many iterations in little CPU time for this class of matrices.

In order to have a clearer idea about the performance of the SPG in practice, we tested it on a set of matrices A taken from the `Matrix Market` repository [18]. The matrix B was considered diagonal with diagonal elements $B_{ii} = i$, $i = 1, \ldots, n$. The numerical results of this experience for a stopping

Table 1 EiCPs solutions with the SPG algorithm

A	Order	Rayleigh			Logarithmic		
		It	T	λ	It	T	λ
Fathy	100	7	0.0012	40.8331	8	0.0015	40.8331
	200	7	0.0051	81.3612	8	0.0053	81.3612
	300	7	0.0231	121.8896	8	0.0111	121.8896
	400	7	0.1343	162.4180	8	0.0197	162.4180
	500	7	0.0272	202.9465	7	0.0271	202.9465
	700	7	0.0536	284.0034	7	0.0536	284.0034
	1,000	7	0.1103	405.5888	7	0.1098	405.5888
pentadiagonal	100	355	0.0106	1.3309	224	0.0059	1.3309
	200	692	0.0387	1.3327	1,004	0.0544	1.3327
	300	1,976	0.1867	1.3330	1,378	0.1130	1.3330
	400	2,811	0.3157	1.3332	2,901	0.3084	1.3332
	500	3,857	0.5451	1.3332	3,146	0.4256	1.3332
	700	5,073	1.1542	1.3333	6,638	1.2813	1.3333
	1,000	9,344	2.7180	1.3333	8,239	2.3036	1.3333
	2,000	17,655	10.8075	1.3333	12,579	7.3733	1.3333
	5,000	16,272	27.7868	1.3333	13,063	18.8608	1.3333
	10,000	13,259	45.2588	1.3333	13,063	37.3628	1.3333
	20,000	12,881	83.2591	1.3333	14,867	92.6165	1.3333

tolerance of $\epsilon = 10^{-6}$ are displayed in Table 2 and show that, as before, the SPG algorithm has been able to find a solution of the EiCP in a very fast way. Furthermore, in general the algorithm has required few iterations to terminate. The performance of the method is the worst for the test problem `nos5`, where, as for the pentadiagonal matrices, the number of iterations required by the algorithm to get an accurate solution of the EiCP is large. These results are not surprising, as the SPG algorithm only uses first order derivative information and may have slow progress in the last iterations. To illustrate this type of behavior, we display the numerical results of the performance of the SPG algorithm for $\epsilon = 10^{-5}$ and $\epsilon = 10^{-4}$ in Table 3. Note that for $\epsilon = 10^{-4}$ the algorithm requires a quarter of the number of iterations that have been performed to get the most accurate solution associated with $\epsilon = 10^{-6}$. It is also interesting to note that the SPG algorithm has found different stationary points for the two merit functions in two examples (`nos4` and `nos6`). This may be

Table 2 EiCPs solutions for `Matrix Market` matrices with the SPG algorithm

A	Order	Rayleigh			Logarithmic		
		It	T	λ	It	T	λ
bcsstk01	48	28	0.0005	2.5920	22	0.0004	2.5920
bcsstk02	66	164	0.0297	16.5555	82	0.0068	16.5555
nos1	237	6	0.0004	118.5386	10	0.0006	118.5386
nos2	957	3	0.0008	478.5097	7	0.0017	478.5097
nos3	960	97	0.0427	210.4494	129	0.0564	210.4494
nos4	100	26	0.0012	50.8760	2	0.0001	25.0000
nos5	468	2,918	0.4766	81.1665	2,809	0.4550	81.1665
nos6	675	12	0.0020	2.3356	2	0.0003	0.3827

Table 3 EiCP solutions for nos5 matrix with SPG method

Rayleigh						Logarithmic					
ϵ	It	λ	ϵ	It	λ	ϵ	It	λ	ϵ	It	λ
10^{-5}	1753	81.1639	10^{-4}	832	80.9779	10^{-5}	1807	81.1645	10^{-4}	771	80.9532

explained by the fact that the gradient of the two functions involve different calculations, and the line search combines two different procedures to move in the same direction.

In Table 4 we report the behavior of the well-known packages MINOS/GAMS and LOQO/AMPL for the solution of the same EiCPs. For these experiments we use the initial solution (3.2). These codes use the same initial solution because it produces the best results for both. These codes were applied to the first and second formulations. For matrices of order greater than 500, MINOS/GAMS was unable to solve them and LOQO/AMPL requires too much time to solve these problems. These methods required much more time to achieve a solution than the SPG algorithm. Furthermore, MINOS/GAMS seems to be a better choice for solving EiCP with both formulations, when A is pentadiagonal, while LOQO/AMPL is a better choice for processing EiCP with the logarithmic formulation, when A belongs to the Fathy collection.

For solving the Quadratic formulation by MINOS we use the following initial solution:

$$x_0 = \frac{1}{\sqrt[2]{B_{11}}}\, e^1,$$

where e^1 is the first vector of canonical basis and B_{11} is the element of matrix B that is in the first line and first column. For LOQO, the initial guess is chosen internally. The results of the experiments obtained with this formulation are presented in Table 5, where (***) is used when the algorithm was unable to solve the EiCP, ‘M’ is the number of major iterations and ‘m’ the number of minor iterations in MINOS/GAMS. As before, MINOS/GAMS and LOQO/AMPL could not solve problems of dimensions greater than 500. The LOQO/AMPL is clearly better than MINOS/GAMS to solve problems with Fathy matrices. The later code was able to solve all the problems with pentadiagonal matrices, while LOQO/AMPL was unable to solve two problems.

These results show that the quadratic formulation has some potential for processing the EiCP. However, traditional algorithms, such as MINOS and LOQO, for solving this convex programs, are not competitive with the SPG algorithm for the two remaining formulations. The possible use of an SPG algorithm for solving the quadratic formulation requires an efficient technique to compute the projection on a convex set defined by the intersection of an ellipsoid with the nonnegative orthant. This is a subject for future research.

Based on these experiments, we claim that the SPG algorithm is a very efficient procedure and compares favorably with commercial codes such as LOQO/AMPL and MINOS/GAMS for processing symmetric eigenvalue complementarity problems by exploiting its reduction to stationary points of suitable

Table 4 Symmetric EiCPs solutions with commercial codes

A	Order	MINOS/GAMS						LOQO/AMPL					
		Rayleigh			Logarithmic			Rayleigh			Logarithmic		
		It	T	λ	It	T	λ	It	T	λ	It	T	λ
Fathy	100	193	1.21	40.8331	192	1.21	40.8331	23	0.25	40.8331	17	0.03	40.8331
	200	390	9.46	81.3612	389	9.42	81.3612	21	1.49	81.3612	17	0.16	81.3612
	300	583	31.02	121.8896	583	30.84	121.8896	20	4.95	121.8896	17	0.46	121.8896
	400	778	72.91	162.4180	782	73.48	162.4180	28	18.46	162.4180	17	1.00	162.4180
	500	972	140.62	202.9465	971	142.36	202.9465	26	35.21	202.9464	19	2.00	202.9464
pentadiagonal	100	176	0.08	1.3327	181	0.09	1.3327	22	0.13	1.3309	42	0.16	1.3309
	200	409	0.40	1.3332	411	0.41	1.3332	57	1.71	1.3327	72	1.61	1.3327
	300	631	0.98	1.3333	662	0.99	1.3333	67	5.80	1.3330	84	5.81	1.3330
	400	884	1.86	1.3333	888	1.87	1.3333	69	12.94	1.3332	262	38.83	1.3332
	500	1,181	3.66	1.3333	1,170	3.26	1.3333	108	36.61	1.3332	350	73.56	1.3333

Table 5 Quadratic formulation program solutions

A	Order	MINOS/GAMS				LOQO/AMPL		
		M	m	T	λ	It	T	λ
Fathy	100	9	166	2.03	40.8331	20	0.06	40.8331
	200	10	276	16.68	81.3612	23	0.33	81.3612
	300	13	398	43.69	121.8896	27	1.14	121.8896
	400	16	519	101.81	162.4180	27	2.51	162.4180
	500	19	640	197.14	202.9465	20	3.04	202.9465
pentadiagonal	100	16	331	0.16	1.3327	770	0.42	1.3327
	200	22	554	0.56	1.3332	150	0.16	1.3332
	300	28	739	1.12	1.3333	***		
	400	32	916	2.24	1.3333	105	0.21	1.3333
	500	36	1,112	3.23	1.3333	***		

merit functions. The algorithm is in general able to find a solution of the EiCP with good precision with a quite small computational effort. For some difficult problems, the algorithm can easily compute a solution with low precision, but may require a large amount of calculations to get an accurate solution. We believe that preconditioning techniques could be designed to improve the quality of the solutions for these difficult problems. This should also be a topic for future investigation.

As far as the formulations are concerned, the logarithmic merit function seems to lead into a better performance for the SPG algorithm. Furthermore, the expression of the Hessian for this function is simpler than for the Rayleigh function and this could be another reason to use the logarithmic function for processing the EiCP by a preconditioned spectral projected gradient algorithm.

Acknowledgement We are indebted to an anonymous referee for helpful suggestions.

References

1. Auchmuty, G.: Unconstrained variational principles for eigenvalues of real symmetric matrices. SIAM J. Math. Analy. **20**, 1186–1207 (1989)
2. Barzilai, J., Borwein, J.M.: Two point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
3. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms. 2nd edn., John Wiley and Sons, New York (1993)
4. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont, Massachusetts (1995)
5. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**, 1196–1211 (2000)
6. Birgin, E.G., Martínez, J.M., Raydan, M.: Algorithm 813: SPG—software for convex-constrained optimization. ACM Trans. Math. Softw. **27**, 340–349 (2001)
7. Brooke, A., Kendrick, D., Meeraus, A., Raman, R.: GAMS a User's Guide. GAMS Development Corporation, Washington (December 1998)
8. Intel Corporation: Intel Fortran Compiler User's Guide. Academic Press (2002)
9. Costa, A.P., Figueiredo, I.N., Júdice, J., Martins, J.A.C.: The directional instability problem in systems with frictional contacts. Comput. Methods Appl. Mech. Eng. **193**, 357–384 (2004)

10. Cottle, R., Pang, J., Stone, R.: The Linear Complementarity Problem. Academic Press, Boston (1992)
11. Dai, Y.H., Fletcher, R.: On the asymptotic behaviour of some new gradient methods. Math. Program. **103**, 541–559 (2005)
12. Dai, Y.H., Liao, L.Z.: R-linear convergence of the Barzilai and Borwein gradient method. IMA J. Numer. Anal. **22**, 1–10 (2002)
13. Fletcher, R.: On the Barzilai–Borwein method. In: Qi, L., Teo, K.L., Yang, X.Q. (eds.) Optimization and Control with Applications, pp. 235–256. Springer, Amsterdam (2005)
14. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: a Modeling Language for Mathematical Programming, 2 edn., Duxbury Press (2002)
15. Golub, G.H., Van Loan, C.F.: Matrix Computations. Third edn., The Johns Hopkins University Press, Baltimore (1996)
16. Júdice, J.J., Pires, F.M.: Solution of large-scale separable strictly convex quadratic programs on the simplex. Linear Algebra Appl. **170**, 214–220 (1992)
17. Júdice, J.J., Ribeiro, I., Sherali, H.: The eigenvalue complementarity problem. Comput. Optim. Appl. **37**, 139–156 (2007)
18. Matrix Market: A visual repository of test data for use in comparative studies of algorithms for numerical linear algebra. See the web site http://math.nist.gov/MatrixMarket/
19. Moungeau, M., Torki, M.: Computing Eigenelements of real symmetric matrices via optimization. Comput. Optim. Appl. **29**(3), 263–287 (2004)
20. Murtagh, B.A., Saunders, M.A.: MINOS 5.1 User Guide. Tech. report, Department of Operations Research, Stanford University (1987)
21. Murty, K.G.: Linear Complementarity, Linear and Nonlinear Programming. Heldermann Verlag, Berlin (1988)
22. Queiroz, M.G., Júdice, J.J., Humes Jr., C.: The symmetric eigenvalue complementarity problem. Math. Comput. **73**, 1849–1863 (2004)
23. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. IMA J. Numer. Anal. **13**, 321–326 (1993)
24. Seeger, A.: Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions, Linear Algebra Appl. **292**, 1–14 (1999)
25. Seeger, A., Torki, M.: On eigenvalues induced by a cone constraint. Linear Algebra Appl. **372**, 181–206 (2003)
26. Seeger, A., Torki, M.: Local minima of quadratic forms on convex cones. J. Glob. Optim. (2008, to appear)
27. Shanno, D., Vanderbei, R.J.: An interior-point method for nonconvex nonlinear programming. Comput. Optim. Appl. **13**, 231–252 (1999)
28. Zhou, Y., Gowda, M.S.: On the Finiteness of the Cone Spectrum of Certain Linear Transformations on Euclidean Jordan Algebras. Working Paper, Department of Mathematics and Statistics, University of Maryland, USA (2007)