Variational Inequality Formulation of the Asymmetric Eigenvalue Complementarity Problem and Its Solution by Means of Gap Functions

Carmo P. Brás*, Masao Fukushima†, Joaquim J. Júdice‡, Silvério S. Rosa§

November 3, 2010; revised March 11, 2011

Abstract: In this paper, the solution of the asymmetric eigenvalue complementarity problem (EiCP) is investigated by means of a variational inequality formulation. This problem is then solved by finding a stationary point of the gap function and the regularized gap function. A nonlinear programming formulation of the EiCP results from the gap function. A hybrid algorithm combining a projection technique and a modified Josephy-Newton method is proposed to solve the EiCP by finding a stationary point of the regularized gap function. Numerical results show that the proposed method can in general solve EiCPs efficiently.

Key Words: complementarity problem, eigenvalue problem, variational inequality problem, gap function, regularized gap function.

Mathematics Subject Classification: 15A18, 65F15, 65K05, 90C33

1 Introduction

The eigenvalue complementarity problem (EiCP) is an important problem arising in mechanics, physics and other areas of applied mathematics [8, 9, 27]. The problem consists of finding a real number λ and a vector $x \in \mathbb{R}^n \backslash \{0\}$ such that

$$\begin{gathered} w = (\lambda B - A)x, \\ w \geqslant 0,\ x \geqslant 0, \\ x^T w = 0, \end{gathered}$$

where $w \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{n \times n}$, and B is positive definite. As discussed in [16, 26, 28, 29], if A and B are both symmetric matrices, then the problem is called the symmetric EiCP

*CMA, Departamento de Matemática, Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal. E-Mail: mb@fct.unl.pt

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

‡Departamento de Matemática da Universidade de Coimbra and Instituto de Telecomunicações, 3001-454 Coimbra, Portugal. E-Mail: joaquim.judice@co.it.pt

§Departamento de Matemática da Universidade da Beira Interior and Instituto de Telecomunicações, 6201-001 Covilhã, Portugal. E-Mail: rosa@ubi.pt

and reduces to finding a stationary point of an appropriate merit function on a structured convex set. Therefore a local optimization algorithm can be used to solve the symmetric EiCP. As discussed in [16], the spectral projected gradient (SPG) method [3, 4] is particularly recommended to solve large-scale symmetric EiCPs in practice. If at least one of the matrices A and B is asymmetric, then this reduction is no longer valid and different approaches have to be employed. A number of non-enumerative efficient algorithms have been introduced for solving the asymmetric EiCP [1, 8, 17, 25, 32]. However, these algorithms are not always able to find a solution of the EiCP. An enumerative method has been introduced in [18] that is able to solve the EiCP by computing a global minimum of a special nonconvex merit function on a convex set. The algorithm usually performs well in practice but its computational effort may be exponential in the dimension of the matrices A and B.

In a search for efficient non-enumerative algorithms to solve the asymmetric EiCP, we exploit in this paper its equivalence to a variational inequality problem (VI) discussed in [18]. It is known that the VI can be solved in many cases by finding a stationary point of a special merit function. In this paper, we investigate the use of the so-called gap function [2, 11] and the regularized gap function [10] for solving the VI equivalent to the EiCP. By exploiting the special structure of the EiCP, we show that finding a stationary point of the resulting gap function is equivalent to finding a global minimum of a nonlinear program with nonlinear constraints. Furthermore the optimal value of this problem is equal to zero. We are able to establish a sufficient condition for a Karush-Kuhn-Tucker (KKT) point of this nonlinear program to be a solution of the EiCP. We also investigate the performance of the local optimizer MINOS to solve the EiCP by exploiting this formulation. The algorithm is not always able to find a solution of the EiCP, but it requires a small amount of effort when it is successful.

As discussed in [10], the regularized gap function is continuously differentiable as long as the underlying VI involves a continuously differentiable mapping. For the VI formulation of the EiCP, a gradient-based method can therefore be used to find a stationary point for this function on the simplex. In this paper, we investigate the performance of the spectral projected gradient (SPG) algorithm [3, 4] for such a goal. This choice was done because of the efficiency of this procedure to deal with the symmetric EiCP [16]. Like other local optimization approaches designed for the asymmetric EiCP, the SPG algorithm is not always able to solve the EiCP. As the computation of the gradients is somewhat involved, we also propose a projection algorithm that is designed in such a way that the gradients of the regularized gap function are used as little as possible. Another improvement of the SPG algorithm consists in finding the search direction by using the so-called modified Josephy-Newton (MJN) method [20, 31]. The computation of a search direction for the MJN algorithm requires not only the computation of the gradient of the regularized gap function but the solution of a special mixed linear complementarity problem (MLCP) by an enumerative algorithm [13]. So each iteration of the MJN method is quite expensive. However, this algorithm converges very fast when the initial point is chosen sufficiently near a solution of the VI. In this paper, a hybrid algorithm combining the good features of these three techniques is introduced. Numerical experiments indicate that this algorithm is not always able to solve the EiCP but can usually find a solution with a relatively small number of iterations.

The remainder of this paper is organized as follows. The formulation of the EiCP as a VI on the simplex is given, and the gap function and the regularized gap function for this VI are discussed in Section 2. Section 3 deals with the nonlinear programming (NLP) formulation of the EiCP. The SPG algorithm for finding a stationary point of the regularized gap function is described in Section 4. The projection algorithm and the MJN algorithm are described in Sections 5 and 6, respectively. A hybrid method of the last two algorithms is suggested in Section 7. Computational experience with these algorithms for solving asymmetric EiCPs is reported in Section 8, and some conclusions are included in the last section.

2 VI Formulation of the Asymmetric EiCP

As stated in [18], the EiCP can be formulated as the following variational inequality problem (VI): Find a vector $x \in \Delta$ such that

$$F(x)^T(y - x) \geqslant 0 \quad \forall y \in \Delta, \tag{1}$$

where the function $F : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$F(x) = \left(\frac{x^T Ax}{x^T Bx} B - A\right) x \tag{2}$$

and the set Δ is the unit simplex in $\mathbb{R}^n$, i.e.,

$$\Delta = \{x \in \mathbb{R}^n : e^T x = 1, x \geqslant 0\}. \tag{3}$$

Therefore, the eigenvalue λ is obtained by

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

Here and throughout, we denote by e the vector whose components are all one.

A variety of algorithms has been developed for finding a solution of the VI [7]. Among them, the optimization approach that makes use of a merit function associated with the VI has turned out to be useful in practice. One of the most popular merit functions is the so-called gap function [2, 11], which is defined by

$$f_0(x) = -\min\{F(x)^T(y - x) : y \in \Delta\}. \tag{4}$$

Note that, for any $x \neq 0$, since $F(x)^T x = 0$, we may write

$$f_0(x) = -\min\{F(x)^T y : y \in \Delta\}.$$

The next result shows that we can use this function to reformulate the VI as an equivalent optimization problem.

Theorem 2.1 [2, 11] *A vector x solves VI* (1) *if and only if x is a global optimal solution of the problem*

$$\begin{array}{ll} \text{Minimize} & f_0(x) \\ \text{subject to} & x \in \Delta \end{array} \tag{5}$$

and $f_0(x) = 0$.

Note that the gap function f_0 is not everywhere continuously differentiable, and hence problem (5) is a nonsmooth optimization problem. The regularized gap function [10] has been introduced to overcome such a drawback and is defined by

$$f_\alpha(x) = -\min\left\{F(x)^T(y-x) + \frac{\alpha}{2}\|y-x\|_2^2 : y \in \Delta\right\},$$

where α is a positive parameter. Again, for any $x \neq 0$, we may write

$$f_\alpha(x) = -\min\left\{F(x)^T y + \frac{\alpha}{2}\|y-x\|_2^2 : y \in \Delta\right\}. \tag{6}$$

The next theorem shows that the regularized gap function f can be used to reformulate the VI as an optimization problem.

Theorem 2.2 [10] *A vector x solves VI* (1) *if and only if x is a global optimal solution of the problem*

$$\begin{array}{ll} \text{Minimize} & f_\alpha(x) \\ \text{subject to} & x \in \Delta \end{array}$$

and $f_\alpha(x) = 0$.

Since the function F defined by (2) is continuously differentiable on the simplex Δ, the regularized gap function is also continuously differentiable on Δ. Therefore, any suitable gradient-based method can be employed to find a stationary point of f_α on Δ.

Particularly we remark that the values of the regularized gap function f_α can be computed by solving the following strictly convex separable quadratic program on the simplex:

$$\begin{array}{ll} \text{Minimize} & F(x)^T y + \dfrac{\alpha}{2}\,\|y-x\|_2^2 \\ \text{subject to} & e^T y = 1 \\ & y \geqslant 0. \end{array} \tag{7}$$

Note that the unique optimal solution of problem (7) is given by

$$y = P_\Delta\left(x - \frac{1}{\alpha}F(x)\right),$$

where $P_\Delta(u)$ denotes the projection of a vector $u \in \mathbb{R}^n$ onto the set Δ.

A number of quite efficient algorithms [12, 14, 24, 30] are available to compute the optimal solution of problem (7). In particular, the block principal pivoting (BPP) algorithm discussed in [15] is very simple and has been shown to be strongly polynomial [14]. In order to understand its main steps, let us write the optimality conditions of the above problem as the following mixed linear complementarity problem (MLCP):

$$\begin{gathered} q + \alpha y + \varphi e = w, \\ e^T y = 1,\ w^T y = 0, \\ w \geqslant 0,\ y \geqslant 0,\ \varphi \in \mathbb{R}, \end{gathered}$$

where $q = F(x) - \alpha x$. Due to the special structure of this MLCP, the BPP algorithm [15] can be used to solve it efficiently. At each iteration of the BPP algorithm, one has a complementary solution satisfying $w_i y_i = 0$, $i = 1, \ldots, n$. Thus there is an index set $G \subseteq \{1, 2, \ldots, n\}$ such that $w_i = 0$ for $i \in G$ and $y_i = 0$ for $i \notin G$. The variables y_i, $i \in G$, are called basic, and the vector consisting of those y_i's is denoted y_G. (A similar notation applies to other vectors as well.) We also let e_G denote the vector of ones with the same dimension as y_G. Since the variable φ is unrestricted in sign, it is always treated as basic. Hence the basic variables y_G and φ satisfy the following equations:

$$\begin{cases} q_G + \alpha y_G + \varphi e_G = 0 \\ (e_G)^T y_G = 1. \end{cases}$$

This system can be rewritten as

$$\begin{cases} \varphi = -\dfrac{1}{|G|}\left(\alpha + (e_G)^T q_G\right) \\[2ex] y_G = -\dfrac{1}{\alpha}(q_G + \varphi e_G), \end{cases}$$

where $|G| = (e_G)^T e_G$ is the number of elements in the set G. Then the steps of the BPP algorithm are stated as follows [14]:

Block Principal Pivoting (BPP) Algorithm

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

Step 1. Compute $\varphi := -\dfrac{1}{|G|}\left(\alpha + (e_G)^T q_G\right)$.

Step 2. Let $H := \{i \in G : (q_i + \varphi)/\alpha > 0\}$. If $H = \emptyset$, then stop. The vector y with

$$y_i = \begin{cases} 0 & \text{if } i \notin G \\ -(q_i + \varphi)/\alpha & \text{if } i \in G \end{cases}$$

is the optimal solution of the quadratic program (7). Otherwise, set $G := G \backslash H$ and return to Step 1.

For any given $x \in \Delta$, once the solution y of problem (7) is computed by the BPP algorithm, the value of the regularized gap function f_α is obtained by

$$f_\alpha(x) = -F(x)^T y - \frac{\alpha}{2}(y - x)^T (y - x).$$

3 NLP Formulation with the Gap Function

The set Δ is nonempty, closed and bounded, and therefore the minimum on the right-hand side of (4) is always attained for any vector x. More specifically, by the special feature of the simplex, the gap function f_0 has the following explicit representation on the set Δ:

$$\begin{aligned} f_0(x) &= -\min_{1 \leqslant i \leqslant n} F_i(x) \\ &= -\min_{1 \leqslant i \leqslant n} \left(\frac{x^T A x}{x^T B x} B_i - A_i \right) x, \end{aligned}$$

where $F_i(x)$, A_i and B_i denote the ith component of $F(x)$, the ith rows of A and B, respectively. As mentioned before, the gap function is not everywhere differentiable. However, by introducing an extra variable $\xi \in \mathbb{R}$, the optimization problem (5) can be rewritten as

$$\begin{array}{ll} \text{Minimize} & \xi \\ \text{subject to} & -\left(\dfrac{x^T A x}{x^T B x} B_i - A_i \right) x \leqslant \xi, \quad i = 1, \ldots, n, \\ & x \in \Delta \end{array}$$

or equivalently

$$\begin{array}{ll} \text{Minimize} & \xi \\ \text{subject to} & e\,\xi + \left(\dfrac{x^T A x}{x^T B x} B - A \right) x \geqslant 0, \\ & x \in \Delta. \end{array} \tag{8}$$

Since B is positive definite, problem (8) may further be rewritten as

$$\begin{array}{lll} \text{NLP:} & \text{Minimize} & \xi \\ & \text{subject to} & e\,\xi x^T B x + \left(x^T A x B - x^T B x A \right) x \geqslant 0, \\ & & x \in \Delta. \end{array} \tag{9}$$

Note that, by a basic property of the gap function, the objective value of NLP (9) is nonnegative for any feasible solution (x, ξ). Moreover, by Theorem 2.1, the following result holds:

Theorem 3.1 *A vector $\bar{x}$ is a solution of the EiCP if and only if $(\bar{x}, \bar{\xi})$ is a global optimal solution of NLP* (9) *with $\bar{\xi} = 0$.*

Since NLP (9) is nonconvex, it is in general difficult to find a global optimal solution. Next, we investigate when a Karush-Kuhn-Tucker (KKT) point of NLP (9) is a solution of the EiCP. Let us write NLP (9) in the form

$$\begin{aligned} \text{Minimize} \quad & f(x, \xi) \\ \text{subject to} \quad & e\,\xi\, x^T Bx + g(x) \geqslant 0, && (10) \\ & e^T x = 1, && (11) \\ & x \geqslant 0, && (12) \end{aligned}$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^n$ are defined by

$$f(x, \xi) = \xi \quad \text{and} \quad g(x) = (x^T AxB - x^T BxA)x,$$

respectively. Furthermore, let $r \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$ and $y \in \mathbb{R}^n$ be the Lagrange multipliers associated with the constraints (10), (11) and (12), respectively. A KKT point for the above problem satisfies

$$\begin{aligned} & \nabla_x f(x, \xi) - \xi e^T r(B + B^T)x - \nabla g(x)r + \alpha e = y, \\ & \nabla_\xi f(x, \xi) - e^T r x^T Bx = 0, && (13) \\ & e\xi x^T Bx + g(x) \geqslant 0,\ r \geqslant 0,\ r^T \left(e\xi x^T Bx + g(x)\right) = 0, && (14) \\ & e^T x = 1, \\ & x \geqslant 0,\ y \geqslant 0,\ x^T y = 0, \end{aligned}$$

with $\nabla_x f(x, \xi) = 0$, $\nabla_\xi f(x, \xi) = 1$ and

$$\nabla g(x) = Bxx^T(A + A^T) - Axx^T(B + B^T) + (x^T Ax)B^T - (x^T Bx)A^T.$$

From (13) we obtain

$$e^T r = \frac{1}{x^T Bx} > 0,$$

i.e., r is a nonzero nonnegative vector. It is now easy to prove the following sufficient condition for a KKT point of NLP (9) to be a solution of the EiCP.

Theorem 3.2 *If the Lagrange multiplier vector r associated with the constraint* (10) *is positive, then a KKT point of NLP* (9) *is a solution of the EiCP.*

Proof: Since $r > 0$, condition (14) implies that

$$e\,\xi\, x^T Bx + g(x) = 0. \tag{15}$$

Multiplying both sides of (15) by x^T, we obtain

$$\xi\, x^T e\, x^T Bx + x^T g(x) = 0. \tag{16}$$

From the definition of g, it is easy to see that $x^T g(x) = 0$. Since $x^T e = 1$ and $x^T Bx > 0$, (16) yields $\xi = 0$. The assertion of the theorem then follows from Theorem 3.1. $\square$

In general this sufficient condition does not always hold and a KKT point of NLP (9) may not be a solution of the EiCP. In Section 8, we report some numerical experience for finding a KKT point of NLP (9) with the local optimizer code MINOS [22].

4 Spectral Projected Gradient Algorithm

In this section, we describe the spectral projected gradient (SPG) method applied to the EiCP. Specifically, the SPG method searches for a stationary point of the regularized gap function f_α on the simplex Δ. At iteration k, given $x_k \in \Delta$, the projected gradient search direction d_k is given by

$$d_k = P_\Delta(x_k - \eta_k \nabla f_\alpha(x_k)) - x_k. \tag{17}$$

Then x_k is updated by $x_{k+1} = x_k + \delta_k d_k$, where the stepsize $\delta_k \in (0, 1]$ is computed by a line search technique to satisfy the Armijo rule:

$$f_\alpha(x_k + \delta_k d_k) \leqslant f_\alpha(x_k) + \delta_k \beta \nabla f_\alpha(x_k)^T d_k, \tag{18}$$

where $0 < \beta < 1$. As discussed in [3], the algorithm converges to a stationary point of f_α under reasonable hypotheses. Next we discuss the main issues in the SPG algorithm when applied to the problem of minimizing the regularized gap function f_α on the simplex Δ.

(i) Computation of the search direction d_k:
By definition, $P_\Delta(u)$ is the unique optimal solution of the following strictly convex separable quadratic programming problem:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}\|y - u\|_2^2 \\ \text{subject to} & e^T y = 1, \\ & y \geqslant 0. \end{array} \tag{19}$$

This problem can effectively be solved using the BPP method discussed in Section 2. By (17), the search direction d_k is obtained by $d_k = \bar{y} - x_k$, where $\bar{y}$ is the optimal solution of problem (19) with $u = x_k - \eta_k \nabla f_\alpha(x_k)$.

(ii) Computation of $\nabla f_\alpha(x)$:
The gradient of the regularized gap function that is required in each iteration is computed by the steps presented below.

(I) After a few algebraic manipulations on the function F defined by (2), we obtain the (transposed) Jacobian matrix $\nabla F(x)$ as

$$\nabla F(x) = \frac{x^T A x}{x^T B x} B^T + \frac{1}{x^T B x}(A + A^T) x x^T B^T - \frac{x^T A x}{(x^T B x)^2}(B + B^T) x x^T B^T - A^T.$$

Let $\lambda = \dfrac{x^T A x}{x^T B x}$ and $v = \dfrac{1}{x^T B x} B x$. Then we can write

$$\nabla F(x) = \lambda B^T + (A + A^T) x v^T - \lambda (B + B^T) x v^T - A^T. \tag{20}$$

(II) As in (i), the projection $P_\Delta(x - \frac{1}{\alpha}F(x))$ can be computed by solving problem (19) with $u = x - \frac{1}{\alpha}F(x)$.

(III) Let $y = P_\Delta(x - \frac{1}{\alpha}F(x))$. Then, noticing that f_α is given by (6), it can be shown from Danskin's theorem [5] that the gradient $\nabla f_\alpha(x)$ is calculated by

$$\nabla f_\alpha(x) = -\nabla F(x) y - \alpha(x - y),$$

where $\nabla F(x)$ is given in (I).

(iii) Computation of the spectral parameter η_k:
Let $w_k = \nabla f_\alpha(x_k)$ and η_{min} and η_{max} be positive real numbers such that $\eta_{min} < \eta_{max}$. Then η_k is computed as follows [4, 16]:

(I) For $k = 0$, let

$$\eta_0 = \text{mid}\left\{\eta_{min},\ \eta_{max},\ \frac{1}{\|P_\Delta(x_0 - w_0) - x_0\|_\infty}\right\},$$

where mid$\{a, b, c\}$ denotes the middle value of three numbers a, b and c.

(II) For any $k > 0$, let

$$s_{k-1} = x_k - x_{k-1}, \quad y_{k-1} = \nabla f_\alpha(x_k) - \nabla f_\alpha(x_{k-1})$$

and

$$\eta_k = \begin{cases} \text{mid}\left\{\eta_{min},\ \eta_{max},\ \dfrac{\langle s_{k-1}, s_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle}\right\} & \text{if } \langle s_{k-1}, y_{k-1}\rangle > \varepsilon \\ \eta_{max} & \text{otherwise,} \end{cases} \tag{21}$$

where ε is a small positive number.

The SPG algorithm is simple structurally and converges to a stationary point of the regularized gap function f_α on Δ. Despite its simplicity, the computation of the gradient of the regularized gap function requires much effort. Furthermore, numerical experience to be presented in Section 8 shows that the algorithm often faces difficulties to terminate and may be slow for some examples. To overcome the first drawback, we introduce in the next section an improvement on the SPG algorithm that tries to use a derivative-free search direction first at each iteration, thereby avoiding the computation of the gradients of f_α as much as possible. To improve the speed of the convergence, on the other hand, we will propose the use of a modified Josephy-Newton algorithm in Section 6.

5 Projection Algorithm

The idea is not to compute the gradient $\nabla f_\alpha(x_k)$ in each iteration, but instead, first try to use the derivative-free search direction given by

$$d_k = P_\Delta\left(x_k - \frac{1}{\alpha}F(x_k)\right) - x_k. \tag{22}$$

Recall that if $d_k = 0$ then x_k is a solution of VI (1) and hence a solution of the EiCP [10]. Furthermore, the following result holds.

Theorem 5.1 *Let $x_k \in \Delta$ and d_k be given by (22). If $d_k \neq 0$, then*

$$F(x_k)^T d_k < 0.$$

Proof: Let $u_k = P_\Delta\left(x_k - \frac{1}{\alpha}F(x_k)\right)$ and $d_k = u_k - x_k \neq 0$. Then u_k is given as the unique optimal solution of the problem

$$\begin{array}{ll} \text{Minimize} & h(y) \\ \text{subject to} & y \in \Delta, \end{array}$$

where the function $h : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$h(y) = \frac{1}{2}\|y - (x_k - \frac{1}{\alpha}F(x_k))\|_2^2.$$

As h is differentiable and Δ is a convex set, u_k satisfies the first-order optimality condition

$$\nabla h(u_k)^T(y - u_k) \geqslant 0 \quad \forall y \in \Delta.$$

Since $x_k \in \Delta$, we have

$$\begin{aligned} \nabla h(u_k)^T(x_k - u_k) \geqslant 0 \quad &\Longleftrightarrow \quad (u_k - (x_k - \frac{1}{\alpha}F(x_k)))^T(x_k - u_k) \geqslant 0 \\ &\Longleftrightarrow \quad (u_k - x_k)^T(x_k - u_k) + \frac{1}{\alpha}F(x_k)^T(x_k - u_k) \geqslant 0 \\ &\Longleftrightarrow \quad \frac{1}{\alpha}F(x_k)^T(x_k - u_k) \geqslant \|u_k - x_k\|_2^2 \\ &\Longrightarrow \quad F(x_k)^T d_k \leqslant -\alpha\|d_k\|_2^2 \end{aligned}$$

and hence the desired result follows. $\square$

In view of this result, we may accept the stepsize δ_k that satisfies a formula similar to the condition (18) with $\nabla f_\alpha(x_k)$ replaced by $F(x_k)$. As in the Armijo rule, δ_k is computed by a finite procedure similar to the one used in the SPG algorithm. If such a stepsize δ_k cannot be obtained after a certain number of trials in the line-search procedure, then the gradient $\nabla f_\alpha(x_k)$ is computed and an usual iteration of the SPG algorithm is performed. The steps of the algorithm are described as follows:

Projection Algorithm (PA)

Step 0. Choose $x_0 \in \Delta$ and a positive integer t_{max} that designates the maximum number of trials allowed in the Armijo line-search for a derivative-free search direction. Choose a constant $\beta \in (0,1)$ and let $k := 0$.

Step 1. Compute $d_k := P_\Delta\left(x_k - \dfrac{1}{\alpha}F(x_k)\right) - x_k$. If $d_k = 0$, terminate. The current vector x_k is a solution of VI (1). Otherwise, use the Armijo rule to find a stepsize $\delta_k \in (0,1]$ satisfying

$$f_\alpha(x_k + \delta_k d_k) \leqslant f_\alpha(x_k) + \delta_k \beta F(x_k)^T d_k. \tag{23}$$

If δ_k is found with the number of trials less than or equal to t_{max}, then go to Step 3. Otherwise, go to Step 2.

Step 2. Compute the gradient $\nabla f_\alpha(x_k)$ and let

$$d_k := P_\Delta(x_k - \eta_k \nabla f_\alpha(x_k)) - x_k,$$

where η_k is the spectral parameter given by (21). If $d_k = 0$, terminate. The current vector x_k is a stationary point of the regularized gap function f_α on Δ. Otherwise, compute a stepsize δ_k satisfying the Armijo rule (18).

Step 3. Update

$$x_{k+1} := x_k + \delta_k d_k$$

and return to Step 1 with $k := k+1$.

It is important to note that the projection in Step 1 can be computed as explained in the previous section. If Step 2 is never visited during the whole procedure, then the algorithm terminates with a solution of VI (1), i.e., a solution of the EiCP. In fact, numerical experience to be reported in Section 8 shows that this algorithm could solve the EiCP in many cases without computing gradients of the regularized gap function. Furthermore, the number of gradients to be computed is usually quite small. This algorithm can therefore be considered an improvement of the SPG method, as the computational effort per iteration is usually much smaller. However, like the SPG method, the projection algorithm is usually slow and may converge to a stationary point of f_α that is not a solution of the EiCP.

6 Modified Josephy-Newton Algorithm

The modified Josephy-Newton (MJN) method solves, at each iteration k, the following affine VI problem AVI(F_k, Δ): Find $x \in \Delta$ such that

$$F_k(x)^T(y - x) \geqslant 0 \quad \forall y \in \Delta, \tag{24}$$

where $F_k : \mathbb{R}^n \to \mathbb{R}^n$ is the linear approximation of the function F at x_k, i.e.,

$$F_k(x) = F(x_k) + \nabla F(x_k)^T(x - x_k),$$

and $\nabla F(x)$ is given by (20). It is known [7] that AVI(F_k, Δ) is equivalent to the following mixed linear complementarity problem (MLCP):

$$\begin{aligned}
&F_k(x) - \lambda e = w,\\
&e^T x = 1,\ \lambda \in \mathbb{R},\\
&x^T w = 0,\ x \geqslant 0,\ w \geqslant 0,
\end{aligned}$$

which is rewritten as

$$\begin{aligned}
&w = q_k + M_k x - \lambda e,\\
&e^T x = 1,\ \lambda \in \mathbb{R},\\
&x^T w = 0,\ x \geqslant 0,\ w \geqslant 0,
\end{aligned} \tag{25}$$

where $q_k = F(x_k) - \nabla F(x_k)^T x_k$ and $M_k = \nabla F(x_k)^T$. As the feasible set Δ is closed, convex and bounded, the AVI(F_k, Δ) always has a solution [7]. Since we cannot expect the matrix M_k to belong to any class of matrices that enjoy certain favorable properties [6], the only approach that is guaranteed to solve MLCP (25) would be an enumerative method. An efficient enumerative method for general MLCPs has been proposed in [13]. This method finds a solution of the MLCP by exploring a binary tree generated by the dichotomy $x_i = 0$ or $w_i = 0$ associated with the complementary condition, see Fig. 1.

In each node of the tree, the algorithm finds a stationary point of the nonconvex quadratic program of the form

$$\begin{array}{ll}
\text{Minimize} & x^T w\\
\text{subject to} & w = q + Mx - \lambda e,\\
& e^T x = 1,\\
& x_i = 0,\ i \in I,\\
& w_j = 0,\ j \in J,\\
& x \geqslant 0,\ w \geqslant 0,
\end{array} \tag{26}$$

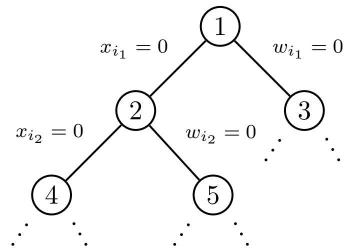

Figure 1: Branching procedure of the enumerative method.

where I and J are the sets defined by $I = \{i : x_i = 0, \text{ fixed}\}$ and $J = \{i : w_i = 0, \text{ fixed}\}$, respectively, in the path of the tree from this node to the root. Furthermore the algorithm contains some heuristic rules for choosing the node and the pair of complementary variables for branching. Numerical results reported in [13] and later in this paper show that the algorithm usually explores very few nodes, i.e., very few quadratic programs of the form (26) are solved, before finding a solution of the MLCP.

The steps of the MJN algorithm are presented below.

Modified Josephy-Newton (MJN) Algorithm

Step 0. Choose $x_0 \in \Delta$ and a small positive number ρ. Let $k := 0$.

Step 1. If $f_\alpha(x_k) = 0$, terminate. The vector x_k is a solution of VI (1), and hence a solution of the EiCP. Otherwise, compute $g_k := P_\Delta(x_k - \eta_k \nabla f_\alpha(x_k)) - x_k$, where η_k is given by (21). If $g_k = 0$, terminate. The vector x_k is a stationary point of the regularized gap function f_α on Δ.

Step 2. Compute $F_k(x)$ and find a solution z_k of AVI(F_k, Δ) by applying the enumerative algorithm to MLCP (25). Let $d_k := z_k - x_k$.

Step 3. If $\nabla f_\alpha(x_k)^T d_k \leq -\rho ||d_k||^2$, then go to Step 4. Otherwise, let $d_k := g_k$.

Step 4. Compute a stepsize $\delta_k \in (0, 1]$ by the Armijo rule (18).

Step 5. Update $x_{k+1} := x_k + \delta_k d_k$ and return to Step 1 with $k := k + 1$.

7 Hybrid Method

In the MJN method, the computational cost per iteration is quite high. In fact not only the gradient of f_α is required but an MLCP has to be solved in each iteration by an enumerative algorithm. On the positive side, the algorithm in general converges quite fast provided the initial point x_0 is close to a solution of the EiCP. This leads to a hybrid method that combines the three techniques discussed in the last three sections. In this procedure, the projection algorithm (PA) described in Section 5 is used by default and a switch to the MJN algorithm incorporating the SPG method is performed when it fails to find a stepsize δ_k satisfying (23) or the value of $f_\alpha(x_k)$ is sufficiently small. Furthermore, the procedure returns to PA if a SPG iteration is performed in Step 3 of the MJN algorithm. In order to describe the steps of the hybrid algorithm, we introduce a zero-one parameter `MJN` that takes the value one when a MJN iteration should be employed and zero otherwise. Moreover, we use a small positive parameter γ to judge if $f_\alpha(x_k)$ is small enough so that we should switch to the MJN algorithm. The steps of the hybrid algorithm are presented below.

Hybrid Algorithm

Step 0. Choose $x_0 \in \Delta$, positive numbers ρ and γ, a positive integer t_{max}, and a constant $\beta \in (0, 1)$. Set `MJN` $:= 0$ and let $k := 0$.

Step 1. If `MJN` $= 1$, go to Step 3. If `MJN` $= 0$ and $f_\alpha(x_k) < \gamma$, set `MJN` $:= 1$ and go to Step 3. Otherwise, go to Step 2.

Step 2. Compute $d_k := P_\Delta\left(x_k - \frac{1}{\alpha}F(x_k)\right) - x_k$. If $d_k = 0$, terminate. The current vector x_k is a solution of VI (1), and hence a solution of the EiCP. Otherwise, use the Armijo rule to find a stepsize $\delta_k \in (0, 1]$ satisfying

$$f_\alpha(x_k + \delta_k d_k) \leqslant f_\alpha(x_k) + \delta_k \beta F(x_k)^T d_k. \tag{27}$$

If δ_k is found with the number of trials less than or equal to t_{max}, then go to Step 7. Otherwise, set `MJN` $:= 1$ and go to Step 3.

Step 3. If $f_\alpha(x_k) = 0$, terminate. The vector x_k is a solution of VI (1), and hence a solution of the EiCP. Otherwise, compute the gradient $\nabla f_\alpha(x_k)$ and let

$$g_k := P_\Delta(x_k - \eta_k \nabla f_\alpha(x_k)) - x_k,$$

where η_k is the spectral parameter given by (21). If $g_k = 0$, terminate. The current vector x_k is a stationary point of the regularized gap function f_α on Δ.

Step 4. Compute $F_k(x)$ and find a solution z_k of AVI(F_k, Δ) by applying an enumerative algorithm to MLCP (25). Let $d_k := z_k - x_k$.

Step 5. If $\nabla f_\alpha(x_k)^T d_k \leq -\rho||d_k||^2$, then go to Step 6. Otherwise, set `MJN` $:= 0$, let $d_k := g_k$ and go to Step 6.

Step 6. Compute a stepsize $\delta_k \in (0, 1]$ by the Armijo rule (18).

Step 7. Update $x_{k+1} := x_k + \delta_k d_k$ and return to Step 1 with $k := k + 1$.

8 Computational Experience

We report some computational experience with the algorithms discussed in the previous sections. All the experiments were carried out using a personal computer with 3.0 GHz Pentium IV processor and 2 GBytes of RAM memory, running Linux 2.6.32. The algorithms were coded in FORTRAN 90 and compiled with the Intel compiler, version 10.0. Running times presented in this section are always given in CPU seconds. The enumerative method required to solve the MLCPs in the MJN algorithm was coded with the active-set method MINOS [22] to solve nonconvex quadratic programs (26). The solver MINOS was also used to solve the nonlinear programming (NLP) formulation of Section 3.

In our first set of test problems, B is always the identity matrix and $A \in \mathbb{R}^{n\times n}$ is an asymmetric matrix from various classes: Lotkin (a modification in the first row of a Hilbert matrix altered to all ones), Murty [23], Tridiagonal, $S \times D$ or $Q \times D$, where S, Q and D are symmetric, orthogonal and diagonal matrices, respectively, or A is randomly generated such that each element is uniformly distributed in the interval $[-1, 1]$. The gallery test matrices of MATLAB [21] were used to generate the orthogonal matrix. The test problems were scaled according to the procedure described in [17], which improves the efficacy of the algorithms, particularly when they are applied to EiCPs with ill-conditioned matrices A. Furthermore, the value of the termination tolerance has been set equal to 10^{-6}. For the SPG algorithm, the values of η_{min} and η_{max} have been fixed to 10^{-3} and 10^{3}, respectively. In the Armijo line search procedure, the stepsize is computed by a finite number of trials $\delta = \dfrac{1}{l^{1.4}}$ for $l = 1, 2, \ldots, t_{max}$ [19]. Furthermore, we let the maximum number of trials be $t_{max} = 10$ in Step 2 of the projection algorithm.

Table 1 includes the computational results with the algorithms for solving asymmetric EiCPs. The initial solution was always chosen at the barycentre of the simplex. In this table, as well as in the sequel, λ is the eigenvalue computed, Ni is the total number of iterations, and T is the total CPU time in seconds spent to solve each problem. We use the notation $\%SPG$ to indicate the percentage of iterations that were performed by the spectral projected gradient direction, and Nin and Nodn denote the average numbers of iterations and nodes, respectively, used in the enumerative method to solve the MLCP in the MJN algorithm. The notations M and m stand for the numbers of *Major* and *minor* iterations, respectively, required by MINOS when applied to the NLP formulation of the EiCP. The symbol (***) indicates that the algorithm was unable to solve a given problem within the maximum of 15000 iterations allowed. For all the problems marked with ‡, the algorithm stopped at a stationary point of the merit function that was not a solution of the VI.

The results show that the SPG algorithm had difficulty in finding a solution to the EiCP. Trouble also occurred with the MJN algorithm for the Murty matrices, where it could not even find a stationary point of the merit function. The algorithms NLP, PA and MJN usually required fewer iterations than the SPG algorithm. However, no algorithm was able to solve all test problems successfully. Furthermore, with the same termination tolerance, the NLP algorithm usually outperforms the other algorithms. The main drawback of the MJN algorithm seems to be the need of using the SPG directions in many iterations. Moreover, the enumerative method solved in most cases the MLCPs required by the MJN algorithm at node 1, that is, without branching.

In Table 2, we display the results of the hybrid algorithm for the same test problems. In the implementation of the hybrid algorithm, the projection method is performed until the value of $f_\alpha(x_k)$ becomes less than or equal to 10^{-3} or a stepsize δ_k that satisfies (27) cannot be obtained within ten trials in the Armijo line search. In these cases, a switch to the MJN algorithm occurs. We use the notations Ni, NiSPG, NiPA and NiMJN to represent the total number of iterations, and the numbers of SPG, PA and MJN iterations, respectively, i.e., Ni=NiSPG+NiPA+NiMJN. In general, the hybrid algorithm performs quite well with the switching value of $\gamma = 10^{-3}$. However, in some cases, this value led to unduly slow convergence and the larger value $\gamma = 10^{-2}$ was used instead. These instances are marked with (*) in Table 2. The results shown in Table 2 demonstrate the efficiency and efficacy of the hybrid algorithm for processing the EiCP, as it generally spent fewer iterations than the other algorithms and was able to solve successfully all but one of the test problems. A further interesting observation is that the number of expensive MJN iterations required to solve these EiCPs was generally small. Furthermore, the number of SPG iterations was always quite small (zero in many cases) in all successful instances. Since PA iterations do not involve much work, the hybrid algorithm is regarded as a fast procedure for solving the EiCP as long as it is not trapped by a stationary point that is not a solution of the problem.

The second set of test problems uses the matrix A as previously defined and a symmetric pentadiagonal strictly diagonally dominant matrix B whose off-diagonal elements $b_{i,\,i-2}$, $b_{i,\,i-1}$ are randomly generated in the interval [0,1] and the diagonal elements $b_{i,\,i}$ are given by

$$b_{i,\,i} = |b_{i,\,i+1}| + |b_{i,\,i+2}| + |b_{i,\,i-1}| + |b_{i,\,i-2}| + 10^{-2}.$$

Hence B is a positive definite matrix [6]. The results with these test problems reported in Table 3 show that the performance of the hybrid algorithm is similar to the previous case where B is the identity matrix.

The computation of multiple solutions of the EiCP has been addressed in [1]. Next, we report two experiments with the projection algorithm (PA) on this issue. In the first experiment, the matrix A has been taken from [1, 25] and is given by

$$A = \begin{bmatrix} 8 & -1 & 4 \\ 3 & 4 & 0.5 \\ 2 & -0.5 & 6 \end{bmatrix}$$

and B is the identity matrix of order 3. By considering 10^5 randomly generated initial points, the PA algorithm was able to find two solutions of this EiCP. Note that this EiCP

Type	n	SPG			NLP				PA				MJN					
		NI	λ	T	M	m	λ	T	NI	λ	$\%SPG$	T	NI	λ	$\%SPG$	NIN	NODN	T
Lotkin	6	15	2.1324	9.00E-04	5	19	2.1324	1.00E-02	17	2.1324	0.0	9.00E-04	3	2.1324	0.0	5.0	1.0	1.35E-02
	10	19	2.4286	1.40E-03	6	37	2.4286	2.00E-02	17	2.4286	0.0	1.10E-03	3	2.4300	0.0	12.3	1.0	2.98E-02
	20	27	2.8065	4.00E-03	6	75	2.8065	1.20E-01	15	2.8065	0.0	2.60E-03	4	2.8071	0.0	22.0	1.0	4.51E-02
	30	29	3.0157	8.10E-03	6	93	3.0157	4.30E-01	16	3.0157	0.0	5.30E-03	4	3.0179	0.0	30.5	1.0	7.76E-02
	40	40	3.1594	1.51E-02	10	160	3.1594	1.35E+00	21	3.1594	0.0	8.60E-03	4	3.1625	0.0	38.2	1.0	1.14E-01
	50	43	‡	2.54E-02	11	196	3.2683	3.03E+00	23	3.2683	0.0	1.33E-02	4	3.2724	0.0	45.8	1.0	1.71E-01
Murty	6	2	0.5000	8.00E-02	27	28	0.5125	1.00E-02	8333	‡	69.4	3.77E-01				***		
	10	9	0.5165	1.40E-03	31	96	0.5893	3.00E-02	396	‡	70.2	3.17E-02				***		
	20	14	‡	2.90E-03	14	86	0.5002	6.00E-02	124	‡	50.0	2.98E-02				***		
	30	16	‡	5.40E-03	17	278	1.3245	4.80E-01	43	‡	39.5	2.16E-02				***		
	40	16	‡	9.10E-03	10	213	0.7085	7.80E-01	65	‡	10.8	3.73E-02				***		
	50	24	‡	1.75E-02	33	487	2.1544	2.86E+00	147	1.9941	6.8	1.73E-01				***		
Tridiagonal	6	55	‡	1.30E-03	5	7	1.3808	1.00E-03	27	1.3808	0.0	9.00E-04	2	1.3811	0.0	8.0	1.0	1.01E-02
	10	164	‡	4.60E-03	5	15	1.3363	1.00E-03	55	1.3363	0.0	1.70E-03	23	1.3387	91.3	14.7	1.0	1.34E-01
	20	608	‡	4.93E-02	11	99	1.3071	5.00E-02	164	1.3071	0.0	7.20E-03	24	1.3132	91.7	52.8	1.0	2.60E-01
	30	1185	‡	2.03E-01	22	399	1.2980	4.40E-01	361	1.2980	0.0	2.58E-02	41	1.2858	97.6	39.1	1.0	5.53E-01
	40	308	‡	9.28E-02	16	193	1.2936	3.50E-01	746	1.2936	0.0	7.97E-02	53	1.3328	92.5	52.0	1.0	1.47E+00
	50	494	‡	2.20E-01	18	288	1.2910	8.40E-01	1492	1.2910	0.0	2.35E-01	256	1.3488	96.9	52.6	1.1	6.95E+00
$S \times D$	6	9	3.4166	9.00E-04	5	5	3.4166	1.00E-02	11	3.4166	0.0	9.00E-04	3	3.4166	0.0	5.0	1.0	1.32E-02
	10	11	‡	1.10E-03	5	10	5.8187	1.00E-02	14	5.8187	0.0	1.30E-03	3	5.8187	0.0	23.0	1.0	2.01E-02
	20	16	‡	2.80E-03	5	20	11.3497	4.00E-02	33	11.3497	0.0	6.10E-03	3	11.3497	0.0	22.7	1.0	3.44E-02
	30	4	‡	3.80E-03	5	31	16.6399	1.40E-01	52	16.6399	0.0	1.75E-02	2	16.6419	0.0	43.5	1.0	4.10E-02
	40	4	‡	6.60E-03	5	42	21.8470	3.60E-01	62	21.8470	0.0	3.82E-02	2	21.8490	0.0	184.0	1.0	7.41E-02
	50	5	‡	9.30E-03	5	52	27.0162	7.90E-01	70	27.0162	0.0	7.40E-02	2	27.0183	0.0	251.6	1.0	1.11E-01
$Q \times D$	6	47	‡	1.20E-03	5	17	0.7949	2.00E-02	12	0.6039	0.0	7.00E-04	3	0.6055	0.0	6.0	1.0	1.32E-02
	10	58	‡	2.20E-03	7	36	0.5639	2.00E-02	19	0.5639	0.0	1.20E-03	5	1.1931	0.0	26.0	1.4	3.40E-02
	20	157	‡	1.44E-02	8	121	0.5326	1.90E-01	42	1.5445	4.8	4.70E-03	49	0.7469	87.8	61.4	3.1	7.22E-01
	30	473	‡	8.24E-02	9	154	0.5219	7.10E-01	159	‡	52.2	8.45E-02	42	1.3839	95.2	125.9	3.7	1.14E+00
	40	933	‡	2.72E-01	9	174	0.5165	1.72E+00	122	‡	59.8	1.18E-01	666	1.6717	97.1	94.6	1.1	2.08E+01
	50	981	‡	4.27E-01	11	280	0.5132	4.83E+00	47	2.9946	2.1	2.25E-02	76	‡	96.1	89.8	2.0	3.18E+00
Uniform $[-1, 1]$	6	17	‡	1.00E-03	8	14	1.0332	1.00E-03	29	-0.0056	34.5	1.50E-03	17	‡	94.1	0.0	1.0	6.28E-02
	10	52	‡	2.20E-03	8	43	‡	2.00E-02	189	0.4823	0.5	3.10E-03	4	0.4840	0.0	14.8	2.5	2.81E-02
	20	254	‡	2.34E-02	12	207	1.7921	2.80E-01	45	1.7921	0.0	4.00E-03	5	1.7921	0.0	24.6	1.0	5.50E-02
	30	312	‡	5.69E-02	13	163	‡	6.40E-01	166	‡	82.5	1.19E-01	7	1.5167	14.3	150.1	5.9	2.17E-01
	40	138	‡	4.42E-02	12	251	‡	2.25E+00	77	3.1125	0.0	2.44E-02	6	3.1132	16.7	120.7	2.2	2.07E-01
	50	518	‡	2.29E-01	12	336	0.7594	4.13E+00	777	‡	76.2	1.36E+00	243	‡	98.8	2895.8	49.4	8.58E+01

Table 1: Performance of SPG, NLP, PA and MJN algorithms for asymmetric EiCPs.

Type	n	Hybrid Algorithm NI	λ	NISPG	NIPA	NIMJN	T
Lotkin	6	5	2.1326	0	4	1	5.80E-03
	10	6	2.4290	0	5	1	6.90E-03
	20	7	2.8083	0	6	1	1.34E-02
	30	8	3.0160	0	6	2	4.17E-02
	40	9	3.1603	0	7	2	6.11E-02
	50	9	3.2730	0	7	2	9.52E-02
Murty	6	8	0.9415	2	5	1	1.69E-02
	10	11	1.4769	3	7	1	2.54E-02
	20	7	2.5171	0	6	1	1.51E-02
	30*	43	3.1564	2	29	12	2.65E-01
	40	23	4.7846	1	8	14	4.64E-01
	50	45	2.1188	2	28	15	6.94E-01
Tridiagonal	6	6	1.3817	0	5	1	5.30E-03
	10	12	1.3410	3	8	1	4.41E-02
	20*	3	1.3094	0	1	2	2.61E-02
	30*	12	1.3101	5	6	1	1.26E-01
	40	6	1.3053	2	3	1	9.58E-02
	50	2	1.3020	0	1	1	5.86E-02
$S \times D$	6	4	3.4166	0	3	1	5.80E-03
	10	5	5.8189	0	4	1	7.00E-03
	20	9	11.3503	0	8	1	1.50E-02
	30	13	16.6410	0	12	1	3.35E-02
	40	16	21.8478	0	15	1	5.74E-02
	50	18	27.0176	0	17	1	8.71E-02
$Q \times D$	6	8	0.9503	1	3	4	2.70E-02
	10	14	0.9291	4	9	1	2.92E-02
	20	10	1.7897	1	7	2	5.13E-02
	30	7	2.2343	1	5	1	4.23E-02
	40	7	2.6359	1	5	1	6.18E-02
	50	7	2.9948	1	5	1	9.32E-02
Uniform $[-1, 1]$	6	11	-0.0048	4	5	2	2.61E-02
	10	10	0.4825	1	7	2	1.85E-02
	20	10	1.7925	0	9	1	1.40E-02
	30	205	‡	184	19	2	3.54E+00
	40	15	3.1129	0	14	1	3.67E-02
	50*	14	2.3404	1	5	8	8.56E-01

Table 2: Performance of the hybrid algorithm for asymmetric EiCPs.

has exactly three eigenvalues that can be obtained by a complete enumeration procedure described in [26]. These results are displayed in Table 4, where "Freq" stands for the frequency of each eigenvalue in percentage.

In the second experiment, B is again the identity matrix and A is a symmetric positive definite matrix of order 10 randomly generated. By using 10^4 randomly generated initial points, the PA algorithm was able to find two of the four solutions of the EiCP with the

Type	n	Hybrid Algorithm NI	λ	NISPG	NIPA	NIMJN	T
Lotkin	6	4	0.7566	0	2	2	7.00E-03
	10	6	0.9823	0	4	2	9.40E-03
	20	8	1.0138	0	6	2	2.13E-02
	30	9	1.4189	0	6	3	4.32E-02
	40	9	1.4181	0	6	3	6.66E-02
	50	10	1.5002	0	7	3	9.47E-02
$S \times D$	6	6	1.1008	0	5	1	4.31E-03
	10	6	1.4915	0	5	1	5.50E-03
	20	10	3.7751	0	9	1	1.15E-02
	30	15	4.8848	0	14	1	2.11E-02
	40	13	6.0397	0	12	1	3.26E-02
	50	18	7.4878	0	17	1	4.73E-02
$Q \times D$	6	7	0.4561	4	2	1	1.56E-02
	10	7	‡	4	2	1	1.70E-02
	20	3	0.2319	0	2	1	1.11E-02
	30	7	0.5582	1	3	3	6.65E-02
	40	29	0.2601	25	3	1	5.58E-01
	50	16	0.3214	12	3	1	5.13E-01
Uniform $[-1, 1]$	6	8	-0.0019	2	5	1	9.50E-03
	10	17	0.3902	7	6	4	6.57E-02
	20	20	0.3907	5	11	4	1.34E-01
	30	218	‡	206	9	3	2.47E+00
	40	21	1.2768	6	13	2	2.82E-01
	50*	15	1.9626	0	13	2	9.77E-02

Table 3: Performance of the hybrid algorithm for the test problems with a nondiagonal matrix B.

λ	Freq (%)
4.0000	40
9.3979	60

Table 4: Computation of multiple eigenvalues for EiCP with matrices A and B of order 3.

λ	Freq (%)
1.2530	30
1.7005	70

Table 5: Computation of multiple eigenvalues for EiCP with matrices A and B of order 10.

corresponding frequency given in Table 5. Although these experiments show that different eigenvalues could be obtained by using randomly generated initial points, such a random generation strategy is clearly insufficient to find all the eigenvalues. This is a topic that needs further research.

9 Conclusions

In this paper we investigate the solution of the EiCP by exploiting its formulation as a VI on the simplex. The use of the gap function for this VI leads to a nonlinear programming (NLP) formulation of the EiCP. A KKT point for the NLP can be found and in many cases solves the EiCP. A hybrid algorithm combining a projection technique with a modified Josephy-Newton method has been introduced for solving the VI by finding a stationary point of the regularized gap function on the simplex. This algorithm is in general able to find efficiently a solution of the EiCP, but may fail in some occasions. We believe that the efficacy of this algorithm for finding a solution of the EiCP and for computing multiple eigenvalues can be improved. This is a topic of our future research.

References

[1] S. Adly and A. Seeger, A nonsmooth algorithm for cone-constrained eigenvalue problems, *Computational Optimization and Applications*, to appear.

[2] A. Auslender, *Optimisation: Méthodes Numériques*, Masson, Paris, 1976.

[3] E.G. Birgin, J.M. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* 10 (2000), 1196–1211.

[4] E.G. Birgin, J.M. Martínez and M. Raydan, Algorithm 813: *SPG* - Software for convex-constrained optimization, *ACM Transactions on Mathematical Software* 27 (2001), 340–349.

[5] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass., 1995.

[6] R.W. Cottle, J.S. Pang and R.E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[7] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems, vol. II*, Springer-Verlag, New York, 2003.

[8] I.N. Figueiredo, J.J. Júdice, J.A.C. Martins and A. Pinto da Costa, A complementarity eigenproblem in the stability of finite-dimensional elastic systems with frictional contact, In: Ferris, M.C., Mangasarian, O.L. and Pang, J.S. (eds), *Complementarity: Applications, Algorithms and Extensions*, Kluwer Academic Publishers, Dordrecht, 1999, pp. 67–83.

[9] I.N. Figueiredo, J.J. Júdice, J.A.C. Martins and A. Pinto da Costa, The directional instability problem in systems with frictional contacts, *Computer Methods in Applied Mechanics and Engineering* 193 (2004), 357–384.

[10] M. Fukushima, Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems, *Mathematical Programming* 53 (1992), 99–110.

[11] D.W. Hearn, The gap function of a convex program, *Operations Research Letters* 1 (1982), 67–71.

[12] R. Helgason, J. Kennington and H. Lall, A polynomially bounded algorithm for a singly constrained quadratic program, *Mathematical Programming* 18 (1980), 338–343.

[13] J.J. Júdice, A. Faustino and I. Ribeiro, On the solution of NP-hard linear complementarity problems, *Top-Sociedad Española de Estadística e Investigación Operativa* 10(1) (2002), 125–145.

[14] J.J. Júdice and F.M. Pires, Solution of large-scale separable strictly convex quadratic programs on the simplex, *Linear Algebra and its Applications* 170 (1992), 214–220.

[15] J.J. Júdice and F.M. Pires, A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems, *Computers and Operations Research* 21(5) (1994), 587–596.

[16] J.J. Júdice, M. Raydan, S. Rosa and S. Santos, On the solution of symmetric eigenvalue complementarity problem by the spectral projected gradient algorithm, *Numerical Algorithms* 47 (2008), 391–407.

[17] J.J. Júdice, H. Sherali and I. Ribeiro, The eigenvalue complementarity problem, *Computational Optimization and Applications* 37 (2007), 139–156.

[18] J.J. Júdice, H. Sherali, I.M. Ribeiro and S. Rosa, On the asymmetric eigenvalue complementarity problem, *Optimization Methods and Software* 24 (2009), 549–586.

[19] W. La Cruz, J.M. Martínez and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Mathematics of Computation* 75 (2006), 1449–1466.

[20] M. Majig and M. Fukushima, Restricted-step Josephy-Newton method for general variational inequalities with polyhedral constraints, *Pacific Journal of Optimization* 6 (2010), 375–390.

[21] C.B. Moler, J.N. Little and S. Bangert, *Matlab User's Guide - The Language of Technical Computing*, The MathWorks, Sherborn, Mass., 2001.

[22] B. Murtagh and A. Saunders, *MINOS 5.0 user's guide*, Technical Report SOL 83-20, Department of Operations Research, Stanford University, 1983.

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

[24] P. Pardalos and N. Kovoor, An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds, *Mathematical Programming* 46 (1990), 321–328.

[25] A. Pinto Da Costa and A. Seeger, Cone-constrained eigenvalue problems: Theory and algorithms, *Computational Optimization and Applications* 45 (2010), 25–57.

[26] M.G. Queiroz, J.J. Júdice and C. Humes Jr., The symmetric eigenvalue complementarity problem, *Mathematics of Computation* 73 (2004), 1849–1863.

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

[28] A. Seeger and M. Torki, On eigenvalues induced by a cone constraint, *Linear Algebra and its Applications* 372 (2003), 181–206.

[29] A. Seeger and M. Torki, Local minima of quadratic forms on convex cones, *Journal of Global Optimization* 44 (2009), 1–28.

[30] S.M. Stefanov, Convex separable minimization subject to bounded variables, *Computational Optimization and Applications* 18 (2001), 27–48.

[31] K. Taji, M. Fukushima and T. Ibaraki, A globally convergent Newton method for solving strongly monotone variational inequalities, *Mathematical Programming* 58 (1993), 369–383.

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