# On the Solution of the Eigenvalue Complementarity Problem by a Line Search Filter-SQP Algorithm

**Qiu Yu, Zhensheng Yu, Yangchen Liu**

University of Shanghai for Science and Technology, Shanghai, China
Email: 707528593@qq.com, zhsh-yu@163.com

## Abstract

In this paper, the Eigenvalue Complementarity Problem (EiCP) with real symmetric matrices is addressed, which appears in the study of contact problem in mechanics. We discuss a quadratic programming formulation to the problem. The resulting problems are nonlinear programs that can be solved by a line search filter-SQP algorithm.

## Keywords

Eigenvalue Complementarity Problem, Nonlinear Programming, Line Search, Filter Method

## 1. Introduction

The Eigenvalue Complementarity Problem (EiCP) appears in the study of static equilibrium states of finite dimensional mechanical systems with unilateral frictional contact [1] [2] and takes the following form:

$$(\text{EiCP}): \text{Find } \lambda > 0, x \neq 0, \text{such that} \begin{cases} \omega = (\lambda B - A)x \\ \omega \geq 0 \\ x \geq 0 \\ \omega^{\mathrm{T}} x = 0 \end{cases} \quad (1)$$

where $B$ is a positive definite matrix. Note that any solution with $\omega = 0$ correspond to a solution of the (GEiCP).

The Generalized Eigenvalue Complementarity Problem (GEiCP) is

$$(\text{GEiCP})_J : \text{Find } \lambda > 0, x \neq 0, \text{ such that } \begin{cases} \omega = (\lambda B - A) x \\ \omega_{\bar{J}} = 0 \\ \omega_J \geq 0 \\ x_J \geq 0 \\ \omega_J^{\mathrm{T}} x_J = 0 \end{cases}$$

where $B$ is a positive definite matrix, $J \subseteq \{1, 2, \cdots, n\}$ is given, and $\bar{J} = \{1, 2, \cdots, n\} \setminus J$. The (EiCP) is clearly the particular case of the $(\text{GEiCP})_J$ with $J = \{1, 2, \cdots, n\}$.

For any solution $(\lambda x, \omega)$, the value $\lambda$ is called a (general) complementary eigenvalue of $(A, B)$, and $x$ is a corresponding (general) complementary eigenvector [3].

In this paper, the symmetric Eigenvalue Complementarity Problem is studied. Some properties of EiCP are derived, including a necessary and sufficient condition for solvability. When this condition is verified, we can easily obtain an initial point. For the design of an initialization algorithm, we discuss a quadratic programming formulation and a line search filter-SQP algorithm is introduced to solve the (EiCP).

## 2. The Symmetric Eigenvalue Complementarity Problem

In this section, the symmetric EiCP is considered, where the matrices A and B are both symmetric. In this case, the EiCP is closely related to the classical Eigenvalue Problem , and some properties are derived.

**Proposition 2.1.** *Proposition The solvability of the* $(\text{GEiCP})_J$ *is in general a NP-complete problem.*

It follows that solving the (GEiCP) is in general a NP-hard problem. Despite this fact, for some classes of matrices the solvability of the corresponding (EiCP) can be answered easily.

Since the set of complementary eigenvectors of a given complementary eigenvalue is a cone, there is no loss of generality in restricting the problem to finding solutions satisfying $\|x\|_2 = 1$, which replaces the constraint $x \neq 0$. In case of the (EiCP) the linear constraint $\|x\|_1 = e^{\mathrm{T}} = 1$, can be considered instead of $\|x\|_2 = 1$, since $x \geq 0$.

When $A$ and $B$ are both symmetric, the (EiCP) is closely related to the classical Eigenvalue Problem [4] [5]. The complementary condition $\omega^{\mathrm{T}} x = 0$ can be rewritten as $x^{\mathrm{T}} (\lambda B x - A x) = 0$. Since $x \neq 0$, and $B$ is positive definite, then $\lambda(x) = \dfrac{x^{\mathrm{T}} A x}{x^{\mathrm{T}} B x}$, where $\lambda(x)$ is the generalized Rayleigh quotient. The gradient of this function is $\nabla \lambda(x) = \dfrac{2}{x^{\mathrm{T}} B x} \left[ A - \lambda(x) B \right] x$ and $\nabla \lambda(x) = 0$ if and only if $\left[ A - \lambda(x) B \right] x = 0$. Analogously to the classical case, equilibrium points of the Rayleigh quotient in the nonnegative orthant with $\lambda(x) > 0$ are solutions of the (EiCP). This is the main result concerning he practical solution of the symmetric (EiCP).

**Proposition 2.2.** (*EiCP*) *is solvable if and only if there exists some* $x \geq 0$ *such that* $x^{\mathrm{T}} A x > 0$.

Proposition 2.2. shows a necessary and sufficient condition for solvability. For the practical solution of the symmetric (EiCP), finding an initial point is a NP-hard problem. But for some class of matrices which an initial solution for the (OEiCP) can be easily obtained.

**Proposition 2.3.** *Suppose that the matrix A satisfies one of the following conditions*

1) $\exists i : A_{ii} > 0$;

2) $\exists i, j : A_{ii} = 0, A_{jj} \leq 0$ and $A_{ij} > 0$;

3) $A \geq 0, A \neq 0$;

4) *A is a S-matrix* ( $\exists x \geq 0 : Ax > 0$ ).

Then a point $\bar{x} \in X$ such that $\bar{x}^{\mathrm{T}} A \bar{x} > 0$ can be easily obtained and the corresponding (EiCP) is solvable.

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

$$(NLP) \begin{cases} \mathrm{Max} & x^{\mathrm{T}} A x \\ \mathrm{s.t} & x^{\mathrm{T}} B x \leq 1 \\ & x \geq 0 \end{cases} \tag{2}$$

**Theorem 2.1.** *If A is strictly copositive and* $\bar{x} \geq 0$ *is a stationary point of* (*QB*), *then the pair* $\bar{x}$, $\bar{\lambda} = \bar{x}^{\mathrm{T}} A \bar{x}$ *is a solution of* (*EiCP*).

# 3. A Line Search Filter-SQP Algorithm

In this section a line search filter-SQP algorithm is introduced to solve the formulation of the previous section. As we all know, sequential Quadratic Programming (SQP) algorithm is one of the most efficient methods for the numerical solution of constrained nonlinear optimization problems [6]. In the previous methods, penalty function is always used as a merit function to test the acceptability of the iterate points. However, there are several difficulties associated with the use of penalty function and in particular with the choice of the penalty parameter. Too low a choice may result in the loss of an optimal solution; on the other hand, too large a choice damps out the effect of the objective function. Fletcher and Leyffer proposed a filter method as an alternative to the traditional merit function for solving the nonlinear optimization [7]. The significant advantage of the filter method is that it does not need to estimate the penalty parameter which could be difficult to obtain. Recently, the filter method was developed in [8] [9].

Considering a suitable continuously differentiable merit function $f(x)$, it is possible to reduce the (NLP) to the following nonlinear program

$$(P) \begin{cases} \mathrm{Min} & f(x) \\ \mathrm{s.t} & g_j(x) \leq 0 \end{cases} \tag{3}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the constraints $c_i : \mathbb{R}^n \to \mathbb{R}$ $j \in I = \{1, 2, \cdots, m\}$ are twice continuously differentiable.

In SQP methods, at each iteration the search direction is generally obtained by solving the subproblem as follows:

$$\text{QP}_1 \begin{cases} \text{Min} & \dfrac{1}{2} d^{\mathrm{T}} B_k d + \nabla f\left(x_k\right)^{\mathrm{T}} d \\ \text{s.t.} & g_j\left(x_k\right) + \nabla g_j\left(x_k\right)^{\mathrm{T}} d \leq 0 \\ & j \in I = \{1, 2, \cdots, m\} \end{cases} \tag{4}$$

However, the previous QP subproblem has a serious shortcoming that constraints in (QP$_1$) may be inconsistent. To overcome this disadvantage, much attention has been paid [10] [11] [12] [13]. In addition, Liu and Li [12] and Liu and Zeng [13] proposed SQP algorithms with cautious update criteria, which can be considered as modifications of the SQP algorithm given in [11]. In our algorithm, stimulated by [10] [11] [12] [13], the quadratic subproblem (QP$_1$) is replaced by the following problem:

$$Q\left(x_k, B_k, b_k\right) \begin{cases} \text{Min} & \dfrac{1}{2} d^{\mathrm{T}} B_k d + \nabla f\left(x_k\right)^{\mathrm{T}} d + b_k t \\ \text{s.t.} & g_j\left(x_k\right) + \nabla g_j\left(x_k\right)^{\mathrm{T}} d \leq te \\ & j \in I, t \geq 0 \end{cases} \tag{5}$$

where $b_k$ is a positive parameter. Clearly, this subproblem is always consistent and is a convex programming if $B_k$ is positive semidefinite. The KKT condition for the subproblem $Q\left(x_k, B_k, b_k\right)$ are

$$\begin{cases} g_j\left(x_k\right) + \nabla g_j\left(x_k\right)^{\mathrm{T}} d_k \leq t_k e\left(j \in I\right), t_k \geq 0 \\ \nabla f\left(x_k\right) + B_k d_k + \displaystyle\sum_{j \in A\left(x_k\right)} \lambda_{k^+}^j \nabla g_j\left(x_k\right) = 0 \\ b_k - \left\| \lambda_{k^+} \right\|_1 - v_k = 0 \\ \lambda_{k^+}^i \left( g_j\left(x_k\right) + \nabla g_j\left(x_k\right)^{\mathrm{T}} d_k - t_k e \right) = 0 \\ v_k t_k = 0 \\ \lambda_{k^+}^i \geq 0, v_k \geq 0 \end{cases} \tag{6}$$

where $\mathcal{A}\left(x_k\right) = \left\{ j : g_j\left(x_k\right) + \nabla g_j\left(x_k\right)^{\mathrm{T}} d_k - t_k e = 0 \right\}$ is the active set at point $d_k$, $\lambda_{k^+}$ which can be used to determine a new estimate $\lambda_{k+1}$ for the next iteration is the Lagrangian multiplier corresponding to subproblem $Q\left(x_k, B_k, b_k\right)$, and $\lambda_{k^+}^j$ is the ith component of $\lambda_{k^+}$.

After $d_k$ has been computed, a step size $\alpha_k \in (0, 1]$ is determined in order to obtain a next iterate $x_{k+1} = x_k + \alpha_k d_k$ and $\lambda_{k+1} = \lambda_k + \alpha_k \left( \lambda_{k^+} - \lambda_k \right)$.

In this paper, the Lagrangian function value instead of the objective function value is used in the filter together with an appropriate infeasibility measure.

$$L\left(x, \lambda\right) = f\left(x\right) - \lambda^{\mathrm{T}} c\left(x\right) \tag{7}$$

and we define the constraints violation function $h\left(x, \lambda\right)$ by

$$h(x,\lambda) = \sum_{j=1}^{m}\left(\max\left\{0,-c_i(x)\right\}\right)^2 + \left(\sum_{j=1}^{m}\lambda^j \min\left\{0,-c_j(x)\right\}\right)^2 \qquad (8)$$

where $\lambda$ is the Lagrangian multiplier corresponding to NLP satisfied at $x$ and $\lambda^j$ is the ith component of $\lambda$. Define $\lambda_k(\alpha_{k,l}) = \lambda_k + \alpha_{k,l}(\lambda_{k^+} - \lambda_k)$.

In the line search filter technique, for fixed constants $\gamma_h, \gamma_L \in (0,1)$, a trial step size $\alpha_{k,l}$ provides sufficient reduction with respect to the current point $x_k$ if

$$h\left(x_k(\alpha_{k,l}), \lambda_k(\alpha_{k,l})\right) \le (1-\gamma_h)h(x_k,\lambda_k)$$
$$\text{or} \quad L\left(x_k(\alpha_{k,l}), \lambda_k(\alpha_{k,l})\right) \le L(x_k,\lambda_k) - \gamma_L h(x_k,\lambda_k) \qquad (9)$$

Similar to the traditional strategy of the filter method, to avoid the convergence to a feasible point but not an optimal solution, we consider the following L-type [14] [15] switching condition:

$$m_k(\alpha_{k,l}) < 0 \text{ and } \left[-m_k(\alpha_{k,l})\right]^{\omega_L}\left[\alpha_{k,l}\right]^{1-\omega_L} > \delta\left[h(x_k,\lambda_k)\right]^{\omega_h} \qquad (10)$$

where $\delta > 0, \omega_h > 0, \omega_L \ge 1$ and

$$m_k(\alpha) = \alpha\left[\nabla g(x_k)^{\mathrm{T}}d_k + \lambda_k^{\mathrm{T}}g(x_k) - (1-\alpha)(\lambda_{k^+} - \lambda_k)^{\mathrm{T}}g(x_k)\right] \qquad (11)$$

then the Armijo-type reduction condition is employed as follows:

$$L\left(x_k(\alpha_{k,l}), \lambda_k(\alpha_{k,l})\right) \le L(x_k,\lambda_k) + \eta_L m_k(\alpha_{k,l}) \qquad (12)$$

where $\eta_L \in \left(0, \dfrac{1}{2}\right)$ is a fixed constant.

For the sake of a simplified notation, the filter is defined in this paper not as a list but as a set $\mathscr{F}_k \subseteq [0,\infty] \times [0,-\infty]$ containing all $(h,L)$ pairs that are prohibited in iteration $k$. We say that a trial point $x_k(\alpha_{k,l})$ is acceptable to the filter if its $(h,L)$ pair does not in $\mathscr{F}_k$. Given a $\mathscr{F}_0$, throughout the algorithm, the filter is then augmented in some iterations after the new iterate point $x_{k+1}$ has been accepted. We use the updating formula

$$\mathscr{F}_{k+1} = \mathscr{F}_k \cup \left\{(h,L): h \ge (1-\gamma_h)h(x_k,\lambda_k)\right.$$
$$\left. \text{and } L \ge L(x_k,\lambda_k) - \gamma_L h(x_k,\lambda_k)\right\} \qquad (13)$$

In the situation where the trial step size is too small to guarantee sufficient reduction as defined by (10), the method switches to a feasibility restoration phrase, whose purpose is to find a new iterate point that satisfies (10) and is also acceptable to the current filter by trying to decrease the constraint violation. In order to detect the situation where no admissible step size can be found and the restoration phase has to be invoked, we approximate a minimum desired step size using linear models of involved functions. For this, we define that

$$\alpha_k^{\min} := \gamma_\alpha * \begin{cases} \min\left\{\gamma_h, \dfrac{\gamma_L h(x_k,\lambda_k)}{-\varphi_k}, \dfrac{\delta\left[h(x_k,\lambda_k)\right]^{\omega_h}}{\left[-\varphi_k\right]^{\omega_L}}\right\} & \text{if } \omega_k < 0 \\ \gamma_h & \text{otherwise} \end{cases}$$

where $\varphi_k = \nabla g\left(x_k\right)^{\mathrm{T}} d_k + \lambda_k^{\mathrm{T}} g\left(x_k\right) - \left|\left(\lambda_{k^+} - \lambda_k\right)^{\mathrm{T}} g\left(x_k\right)\right|$, $\gamma_\alpha \in (0,1]$, $\delta, \omega_h, \omega_L$ are form (11) and $\gamma_h, \gamma_L$ are form (10). The proposed algorithm switches to the feasibility restoration phase when $\alpha_{k,l}$ becomes smaller than $\alpha_k^{\min}$.

Now, the algorithm for solving the inequality constrained optimization problem (NLP) can be stated as follows:

### Algorithm

Step 1. Initialization: choose an initial point $x_0 \in \mathbb{R}^n$, an initial filter $\mathscr{F}_0 : \left\{\left(h, L\right) : L \geq \tau > h\left(x_0, \lambda_0\right)\right\}$, an initial parameter $b_0 > 0$, a symmetric and positive definite matrix $B_0 \in \mathbb{R}^{n \times n}$, $\delta > 0$, $\delta_1, \delta_2 > 0$, and $\gamma_H, \gamma_L \in (0,1)$. Choose $0 \leq \tau_1 \leq \tau_2 < 1$, $\gamma_\alpha \in (0,1]$, and $\omega_h > 0, \omega_L \geq 1$.

Step 2. Solve the subproblem $Q\left(x_k, B_k, b_k\right)$ to obtain the solution $\left(d_k, t_k\right)$, and let $\left(\lambda_{k^+}, v_k\right)$ be the Lagrange multiplier. If $d_k = 0$ and $t_k = 0$, stop.

Step 3. If $d_k = 0$ but $t_k \neq 0$, set $x_{k+1} = x_k$ and $\mathscr{F}_{k+1} = \mathscr{F}_k$, go to Step 7.

Step 4. If $d_k \neq 0$ and $\varphi_k \geq 0$, go to the feasible restoration phrase in Step 9.

Step 5. If $d_k \neq 0$ and $\varphi_k < 0$, using backtracking line search consider the following:

Step 5.1. Initial line search: set $\alpha_{k,l} = 1$ and $l = 0$, compute the $\alpha_k^{\min}$.

Step 5.2. Compute a new trial point. If the trial step size $\alpha_{k,l} < \alpha_k^{\min}$, go to Step 9. Otherwise, compute new trial point $x_k\left(\alpha_{k,l}\right) = x_k + \alpha_{k,l} d_k$ and $\lambda_k\left(\alpha_{k,l}\right) = \lambda_k + \alpha_{k,l}\left(\lambda_{k^+} - \lambda_k\right)$. Check acceptability to the filter: if the trial point $\left(h\left(x_k\left(\alpha_{k,l}\right)\right), L\left(x_k\left(\alpha_{k,l}\right)\right)\right) \in \mathscr{F}_k$ reject the trial step size and go to Step 5.4.

Step 5.3. Check sufficient decrease with respect to current iterate point:

Step 5.3.1. Case I: condition (11) holds. If the Armijo condition (12) holds, accept the trial step (that is, an L-type iteration), set $x_{k+1} = x_k\left(\alpha_{k,l}\right)$, $\lambda_{k+1} = \lambda_k\left(\alpha_{k,l}\right)$, $\mathscr{F}_{k+1} = \mathscr{F}_k$ and go to Step 5.5; otherwise, go to Step 5.4.

Step 5.3.2. Case II: condition (11) does not hold. If (9) holds, accept the trial step , and go to Step 5.5; otherwise, go to Step 5.4.

Step 5.4. Choose a new trial size $\alpha_{k,l+1} \in \left[\tau_1 \alpha_{k,l}, \tau_2 \alpha_{k,l}\right]$. Set $l = l + 1$, and go back to Step 5.2.

Step 5.5. Accept trial point. Set $x_{k+1} = x_k\left(\alpha_{k,l}\right)$ and $\alpha_k = \alpha_{k,l}$.

Step 6. Augment filter if it is necessary. If $k$ is not an L-type iteration, augment the filter using (13); otherwise leave the filter unchanged; that is, set $\mathscr{F}_{k+1} = \mathscr{F}_k$.

Step 7. Update parameters. Compute $\alpha_k$ by

$$\alpha_k = \min\left\{\left\|d_k\right\|^{-1}, \left\|\lambda_k\right\|_1 + \delta_1\right\}$$

Set

$$b_{k+1} = \begin{cases} b_k & \text{if } b_k \geq \alpha_k \\ b_k + \delta_2 & \text{otherwise} \end{cases}$$

Step 8. Update $B_k$ to $B_{k+1}$. Go to Step 2 with $k$ replaced by $k+1$.

Step 9. Obtain a new point $x_{k+1}$ from a feasible restoration phrase. Set $k = k + 1$, and go to Step 2.

Remark 1. The mechanisms of the filter could ensure that

$\left( h\left( x_k, \lambda_k \right), L\left( x_k, \lambda_k \right) \right) \notin \mathscr{T}_k$ .

Remark 2. The feasibility restoration phrase in Step 9 could be any iterative algorithm with the goal of finding a less infeasible point.

Remark 3. Step 3 - Step 7 which are call the inner loop are backtracking line search whose goal is to find the $\alpha_k$ such that $x_k + \alpha_k d_k$ is acceptable as the next iteration.

Remark 4. In Step 9, we decrease $h(x, \lambda)$ to get $x_{k+1}$ by fixing $\lambda = \lambda_k$ .

## 4. Global Convergence of Algorithm

In this section, we will show the proposed algorithm is well defined and globally convergent under some mild donditions. Let $\bar{x}$ be a feasible point of NLP (1) and $\bar{\lambda}$ be the multiplies corresponding to the nonlinear constraints satisfied at $\bar{x}$. Denote the *i*th component of $\bar{\lambda}$ by $\bar{\lambda}^i$. Then throughout this paper, we always assume that them following assumptions hold:

### Assumptions

1) The function $L(x, \lambda)$ and $g(x)$ are twice continuously differentiable.

2) The sequence $\{x_k\}$ remains in a compact and convex subset $X \in \mathbb{R}^n$.

3) The strict Mangasarian-Fromowitz constraint qualification (SMFCQ) condition holds when $\bar{x}$ is a feasible point of the NLP. Then for $\bar{x}$, the vectors $\{\nabla g_j(x), j \in I(x)\}$ are linearly independent, where $I(x) = \{j \in I \mid g_j(x) = h(x, \lambda)\}$. There exists a constant $M_B > 0$ such that $z^\mathrm{T} B_k z \geq M_B \|z\|^2$, $z \in \{z : \nabla g_j(x_k)^\mathrm{T} z = 0, j \in I(x)\}$. And there exist two constants $q \geq p > 0$ such that the matrices sequence $\{B_k\}$ satisfy $p\|d\|^2 \leq d^\mathrm{T} B_k d \leq q\|d\|^2$ for all $k$ and $d \in \mathbb{R}^n$.

4) $d_k, \lambda_k$ are uniformly bounded, *i.e.*, there exist constants $M_d > 0$ and $M_\lambda > 0$, such that $\|d_k\| \leq M_d$, $\|\lambda_k\|_\infty \leq M_\lambda$.

In the following, we first show that under Assumptions G the sequence of infeasibility measure $\{h(x_k)\}$ converges to zero, and all limit points of $\{x_k\}$ are feasible.

**Lemma 4.1.** *Suppose that Assumptions hold and that the filter is augmented only finite number of times. Then* $\lim\limits_{k \to +\infty} h(x_k, \lambda_k) = 0$ .

**Lemma 4.2.** *Suppose that Assumptions hold and let* $\{x_k\}$ *be the sequence of iterates generated by Algorithm so that the filter is augmented in every iteration k. Then* $\lim\limits_{k \to +\infty} h(x_k, \lambda_k) = 0$ .

The proof of the Lemma 4.1 and Lemma 4.2 are similar to Lemma 5 in [14] by replacing $f(x)$ by $L(x, \lambda)$, we omit it. After we have the previous two lemmas, we can get the following theorem. That is to say that the limit point of $\{x_k\}$ is feasible. The proof of the next theorem is similar to the Theorem 1 in [14].

**Theorem 4.1.** *Suppose that Assumptions G hold. Then* $\lim\limits_{k \to +\infty} h(x_k, \lambda_k) = 0$ .

**Lemma 4.3.** *Suppose that Assumptions hold and let* $\bar{x}$ *be a feasible point of NLP at which SMFCQ condition holds, but not a KKT point. If* $\{x_k\}$ *is a subsequence of iterate points for which* $\chi(x_k) \geq 0$ *with a constant* $\epsilon > 0$ *, then*

*there exist constants* $\epsilon_1 > 0$, $\epsilon_2 > 0$, *such that*

$$h(x_k, \lambda_k) \le \epsilon_1 \Rightarrow m_k(\alpha) \le -\epsilon_2 \alpha$$

**Lemma 4.3.** shows that the search direction, the optimal solution of the subproblem is a sufficient descent direction for Lagrangian function at points that are nonoptimal and sufficiently close to feasible point. And the next lemma shows that no $(h, L)$-pair corresponding to a feasible point is ever including in the filter.

**Lemma 4.4.** *Suppose that Assumptions hold. Then*

$\Omega_k := \min\{h : (h, L) \in \mathscr{F}_k\} > 0$   *for all k.*

**Lemma 4.5.** *Suppose that Assumptions hold. Let* $\{x_{k_i}\}$ *be a subsequence with* $m_{k_i}(\alpha) \le -\epsilon_2 \alpha$ *for a constant* $\epsilon_2 > 0$ *independent of* $k_i$ *and for all* $\alpha \in (0, 1]$. *Then there exists a constant* $\bar{\alpha} > 0$ *so that for all* $k_i$ *and* $\alpha < \bar{\alpha}$, *we have*

$$L\left(x_{k_i} + \alpha d_{k_i}, \lambda_{k_i} + \alpha\left(\lambda_{k_i}^+ - \lambda_{k_i}\right)\right) - L\left(x_{k_i}, \lambda_{k_i}\right) \le \eta_L m_{k_i}(\alpha)$$

The Lemma 4.5 shows that there exists a step length bounded away from zero so that the Armijo condition for the Lagrangian function is satisfied under some conditions. The next lemma shows that there is no cycle between Step 3 and Step 7 in above algorithm. The proof of the Lemma 4.6 is similar to Lemma 4 in [16].

**Lemma 4.6.** *Suppose that Assumptions G hold. Then the inner loop terminates in a finite number of iterations.*

**Lemma 4.7.** *Suppose that the filter is augmented only a finite number of times; that is,* $|\mathscr{L}| < \infty$. *Then* $\lim_{k \to +\infty} d_k = 0$.

There the proof is stated for slightly different circumstances, but it is easy to verify that it is still valid in our context. And we now show the global convergence result under some mild conditions. The proof of the next two theorems is similar to the Theorem 10 and Theorem 11 in [17].

**Theorem 4.2.** *Suppose that all stated assumptions hold. Furthermore, assume that* $\{x_k\}$ *is an infinite sequence generated by Algorithm and* $|\mathscr{L}| < \infty$. *Then every limit point is a KKT point.*

**Theorem 4.3.** *Suppose that all stated assumptions hold. Furthermore, assume that* $\{x_k\}$ *is an infinite sequence generated by Algorithm and* $|\mathscr{L}| = \infty$. *Then there exists at least one accumulation point which is a KKT point.*

## 5. Numerical Experiments

In this section, some computational experience is presented to illustrate the efficiency of the algorithms described in this paper for the solution of symmetric EiCP. All programs are written in MATLAB and run on a Dell version 4510U with a 2.8 GHz Intel i7 processor. For our test problems, the matrix $B = I_n$, and the matrices $A \in \mathbb{R}^{n \times n}$ were randomly generated. The test problems are scaled according to the procedure described in [18]. 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.

As described in Proposition 2.3, the initial solution $x_0$ can be chosen by one several processes. In particular if $A$ has at least one diagonal element $\alpha_{ii} > 0$ then the initial solution can be chosen as $x_0 = pe^i$, where $e^i$ is the vector $i$ of the canonical basis.

The parameters used in the algorithm are follows: $B_0 = E_n$, $b_0 = 111$, $\delta = 1$, $\delta_1 = 1$, $\delta_2 = 1$, $\gamma_H = \gamma_L = 0.5$, $\gamma_\alpha = 0.5$, $\tau_1 = \tau_2 = 0.5$, $\omega_H = \frac{1}{3}$, $\omega_L = 2$, $\varepsilon = 10^{-6}$. The stop criteria are $\left\| d_0^k \right\|$ sufficiently small, where $d_0^k = \left( d_0^{\mathrm{T}}, t_k \right)^{\mathrm{T}}$. In particular, the stop criteria of Step 2 are changed to: if $\left\| d_0^k \right\| \leq 10^{-6}$, stop. We update the matrices $B_k$ using the BFGS formula and use the matlab toolbox to solve the subproblem $Q(x_k, B_k, b_k)$.

The test results are given in Table, the notations mean follows: No: the number of the problems, $m$: the dimension of the vector; T: the total CPU time in seconds for solving the problem; OPT: the final value of the objective function; $\lambda$: the eigenvalue of the EiCP.

| No | $m$ | $T$ | OPT | $\lambda$ |
|---|---|---|---|---|
| 1 | 3 | 0.8807 | $(0.9256, 0.3799, 0.0000)^{\mathrm{T}}$ | 3.3097 |
| 2 | 3 | 0.8645 | $(0.7833, 0.6216, 0.0000)^{\mathrm{T}}$ | 36.9029 |
| 3 | 3 | 0.7931 | $(0.7947, 0.0000, 0.6070)^{\mathrm{T}}$ | 26.4018 |
| 4 | 4 | 1.8999 | $(0.9125, 0.3960, 0.0000, 0.1024)^{\mathrm{T}}$ | 5.8678 |
| 5 | 4 | 0.0220 | $(0.4472, 0.4472, 0.4472, 0.4472)^{\mathrm{T}}$ | 6.0000 |
| 6 | 4 | 1.9396 | $(0.9886, 0.0000, 0.0000, 0.1508)^{\mathrm{T}}$ | 69.4806 |
| 7 | 5 | 1.9118 | $(0.4653, 0.3686, 0.4639, 0.3818, 0.5354)^{\mathrm{T}}$ | 26.6925 |

Based on the results, we claim that this algorithm is a efficient procedure to solve Symmetric Eigenvalue Complementarity Problems.

## 6. Conclusion

In this paper, the Eigenvalue Complementarity Problem with real symmetric matrices have been considered, and we reformulate the EiCP to a NLP and use a line search filter-SQP algorithm to solve them. The numerical experiments show that this method is a promising method for solving the EiCP. Actually, we can solve the problem as a quadratically constrained quadratic programming problem not a NLP. And the design of efficient procedure for solving the quadratically constrained quadratic programming subproblems is also one of our current research areas.

## References

[1] Costa, A., Figueiredo, I., Júdice, J. and Martins, J. (2001) A Complementarity Eigen Problem in the Stability Analysis of Finite Dimensional Elastic Systems with Fric-

tional Contact. *Complementarity: Applications, Algorithms and Extensions,* **50**, 67-83.

[2]    Costa, A., Martins, J., Figueiredo, I. and Júdice, J. (2004) The Directional Instability Problem in Systems with Frictional Contacts. *Computer Methods in Applied Mechanics and Engineering*, **193**, 357-384.

[3]    Humes, C., Júdice, J. and Queiroz, M. (2004) The Symmetric Eigenvalue Complementarity Problem. *Mathematics of Computation*, **73**, 1849-1863.

[4]    Júdice, J., Sherali, H. and Ribeiro, I. (2007) The Eigenvalue Complementarity Problem. *Computational Optimization and Applications*, **37**, 139-156.
https://doi.org/10.1007/s10589-007-9017-0

[5]    Parlett, B. (1997) The Symmetric Eigenvalue Problem. Society for Industrial and Applied Mathematics, University of California, Berkeley, California.

[6]    Boggs, P. and Tolle, J. (1995) Sequential Quadratic Programming. *Acta Numerica*, **4**, 1-51. https://doi.org/10.1017/S0962492900002518

[7]    Fletcher, R. and Leyffer, S. (2002) Nonlinear Programming without a Penalty Function. *Math. Program*, **91**, 239-269. https://doi.org/10.1007/s101070100244

[8]    Gould, N.I.M., Loh, Y. and Robinson, D.P. (2014) A Filter Method with Unified Step Computation for Nonlinear Optimization. *The SIAM Journal on Optimization*, **24**, 175-209. https://doi.org/10.1137/130920599

[9]    Milzarek, A. and Ulbrich, M. (2014) A Semismooth Newton Method with Multidimensional Filter Globalization for $L1$-Optimization. *The SIAM Journal on Optimization*, **24**, 298-333. https://doi.org/10.1137/120892167

[10]   Zhang, J.-L. and Zhang, X.-S. (2001) A Modified SQP Method with Nonmonotone Linesearch Technique. *Journal of Global Optimization*, **21**, 201-218.
https://doi.org/10.1023/A:1011942228555

[11]   Pantoja, J.F.A.D.O. and Mayne, D.Q. (1991) Exact Penalty Function Algorithm with Simple Updating of the Penalty Parameter. *Journal of Optimization Theory and Applications*, **69**, 441-467. https://doi.org/10.1007/BF00940684

[12]   Liu, T. and Li, D. (2007) Convergence of the BFGS-SQP Method for Degenerate Problems. *Numerical Functional Analysis and Optimization*, **28**, 927-944.
https://doi.org/10.1080/01630560701405002

[13]   Liu, T.-W. and Zeng, J.-P. (2009) An SQP Algorithm with Cautious Updating Criteria for Nonlinear Degenerate Problems. *Acta Mathematicae Applicatae Sinica*, **25**, 33-42. https://doi.org/10.1007/s10255-004-4100-0

[14]   Wächter, A. and Biegler, L.T. (2005) Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM Journal on Optimization*, **16**, 1-31. https://doi.org/10.1137/S1052623403426556

[15]   Wächter, A. and Biegler, L.T. (2005) Line Search Filter Methods for Nonlinear Programming: Local Convergence. *SIAM Journal on Optimization*, **16**, 32-48.
https://doi.org/10.1137/S1052623403426544

[16]   Jin, Z. (2013) A Globally Convergent Line Search Filter SQP Method for Inequality Constrained Optimization. *Journal of Applied Mathematics*, **2013**, Article ID: 524539.
https://doi.org/10.1155/2013/524539

[17]   Pang, L.-L. and Zhu, D.-T. (2016) A Line Search Filter-SQP Method with Lagrangian Function for Nonlinear Inequality Constrained Optimization. *Japan Journal of Industrial and Applied Mathematics*, **34**, 141-176.
https://doi.org/10.1007/s13160-017-0236-1

[18] Júdice, J., Raydan, M., Rosa, S. and Santos, S. (2008) On the Solution of the Symmetric Eigenvalue Complementarity Problem by the Spectral Projected Gradient Algorithm. *Numerical Algorithms*, **45**, 391-407.
https://doi.org/10.1007/s11075-008-9194-7