Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 13, No. 3, pp. 475–500, July 2017

THE SECOND-ORDER CONE QUADRATIC EIGENVALUE COMPLEMENTARITY PROBLEM

Alfredo N. Iusem*, Joaquim J. Júdice†, Valentina Sessa‡
and Hanif D. Sherali

Abstract: We investigate the solution of the Second-Order Cone Quadratic Eigenvalue Complementarity Problem (SOCQEiCP), which has a solution under reasonable assumptions on the matrices included in its definition. A Nonlinear Programming Problem (NLP) formulation of the SOCQEiCP is introduced. A necessary and sufficient condition for a stationary point (SP) of NLP to be a solution of SOCQEiCP is established. This condition indicates that, in many cases, the computation of a single SP of NLP is sufficient for solving SOCQEiCP. In order to compute a global minimum of NLP for the general case, we develop an enumerative method based on the Reformulation-Linearization Technique and prove its convergence. For computational effectiveness, we also introduce a hybrid method that combines the enumerative algorithm and a semi-smooth Newton method. Computational experience on the solution of a set of test problems demonstrates the efficacy of the proposed hybrid method for solving SOCQEiCP.

Key words: *eigenvalue problems, complementarity problems, nonlinear programming, global optimization, reformulation-linearization technique*

Mathematics Subject Classification: *60F15, 90C33, 90C30, 90C26*

1 Introduction

The *Eigenvalue Complementarity Problem* (EiCP) [29, 31] consists of finding a real number λ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$w = \lambda Bx - Cx \tag{1.1}$$

$$x \geq 0, w \geq 0 \tag{1.2}$$

$$x^\top w = 0, \tag{1.3}$$

where $w \in R^n$, and B and $C \in \mathbb{R}^{n\times n}$, and where B is assumed to be positive definite (PD). This problem finds many applications in engineering [21, 27, 31]. If a triplet (λ, x, w) solves EiCP, then the scalar λ is called a *complementary eigenvalue* and x is a *complementary eigenvector* associated with λ. The condition $x^\top w = 0$ and the nonnegativity requirements on x and w imply that $x_i = 0$ or $w_i = 0$ for all $1 \leq i \leq n$, and so, these pairs of variables are called complementary. The EiCP always has a solution provided that the matrix B is

[*] Alfredo N. Iusem was partially supported by CNPq grant no. 301280/86.

[†] Joaquim J. Júdice was financed by the applicable financial framework (FCT/MEC through national funds and when applicable co-funded by FEDER PT2020 partnership agreement).

[‡] Valentina Sessa was supported by CNPq grant no. 150606/2014-1.

PD [21]. A number of techniques have been proposed for solving EiCP and its extensions [2,6,14,15,19–22,25,28,33].

An extension of the EiCP, called the *Quadratic Eigenvalue Complementarity Problem* (QEiCP), was introduced in [32]. This problem differs from the EiCP through the existence of an additional quadratic term in λ, and consists of finding a real number λ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$w = \lambda^2 Ax + \lambda Bx + Cx \tag{1.4}$$

$$x \geq 0, w \geq 0 \tag{1.5}$$

$$x^\top w = 0, \tag{1.6}$$

where $w \in \mathbb{R}^n$ and A, B, and $C \in \mathbb{R}^{n \times n}$. (We differ from (1.1) and use $+Cx$ in (1.4) for notational convenience.) The λ-component of a solution to QEiCP(A, B, C) is called a *quadratic complementary eigenvalue* and the corresponding x-component is called a *quadratic complementary eigenvector* associated with λ. Contrary to EiCP, QEiCP may have no solution. However, under some not too restrictive conditions on the problem matrices A, B or C, QEiCP always has a solution [4,32], which can be found by either solving QEiCP directly [2,14,15] or by reducing it to a $2n$-dimensional EiCP [4,18]. In particular, semismooth Newton methods [2], enumerative algorithms [14,15,18], and a hybrid method that combines both previous techniques [15,18], have been recommended for solving QEiCP.

The EiCP and the QEiCP can be viewed as mixed nonlinear complementarity problems [10], where the complementary vectors x and w belong to the cone $K = \mathbb{R}^n_+$ and its dual $K^* = \mathbb{R}^n_+$, respectively. The case of EiCP with K being the so-called second-order cone, or Lorentz cone, denoted SOCEiCP, was introduced in [1], and can be stated as follows: Find a real number λ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$w = \lambda Bx - Cx \tag{1.7}$$

$$x \in K, w \in K^* \tag{1.8}$$

$$x^\top w = 0, \tag{1.9}$$

where B and $C \in \mathbb{R}^{n \times n}$, B is PD, and K is the second-order cone defined by

$$K = K_1 \times K_2 \times \ldots \times K_r, \tag{1.10}$$

where

$$K_i = \{x^i \in \mathbb{R}^{n_i} : ||\bar{x}_i|| \leq x^i_0\} \subseteq \mathbb{R}^{n_i},\ (1 \leq i \leq r) \tag{1.11}$$

$$\sum_{i=1}^{r} n_i = n, \tag{1.12}$$

and where

$$x = (x^1, \ldots, x^r) \in \mathbb{R}^n, \tag{1.13}$$

with

$$x^i = (x^i_0, \bar{x}^i) \in \mathbb{R} \times \mathbb{R}^{n_i - 1}, (1 \leq i \leq r). \tag{1.14}$$

Here, $\|\cdot\|$ denotes the Euclidean norm and the dual cone K^* of K is defined by

$$K^* = \{y \in \mathbb{R}^n : y^\top x \geq 0, \forall x \in K\}. \tag{1.15}$$

Observe that each cone K_i is pointed and self-dual, i.e., it satisfies $K_i = K_i^*$. It is well known that the Euclidean Jordan algebra [11] may be associated to the cone K_i. Based on the spectral factorization of vectors in $\mathbb{R}^n$ specified by the Jordan algebra, several approaches that reformulate second-order cone constraints as smooth functions are presented in [16,17]. Similar techniques were analyzed in [1] for finding a solution to SOCEiCP. However, none of these semi-smooth Newton type algorithms in [1] induces global convergence and there is no guarantee that they find a solution to the SOCEiCP even if a line-search procedure is employed.

Alternative approaches for solving SOCEiCP consist of considering this problem as a nonlinear programming problem (NLP) with a nonconvex objective function minimized over a convex set defined by the intersection of the Lorentz cone with a set defined by linear constraints. In the so-called symmetric case (where B and C are both symmetric matrices), the computation of a single stationary point (SP) of this NLP is sufficient to solve the SOCEiCP [5]. In general, the computation of just one SP of NLP may not be enough to find a solution for the SOCEiCP and a global minimum of NLP has to be computed. An enumerative algorithm was introduced in [12] for finding such a global minimum, which explores a binary tree that is constructed by partitioning the interval associated with some selected components of the complementary variables involved. Whereas the NLP can be rewritten using a cone-wise formalism, in [12] and in this paper, a component-wise notation is adopted, because it is consistent with the mode of operation of the enumerative method. A hybrid algorithm combining this enumerative method and a semi-smooth Newton algorithm was also introduced in [12] to enhance the computational efficiency of the enumerative method.

Encouraged by the good results obtained by applying the hybrid method for solving SOCEiCP, we investigate in this paper a similar method for solving the Second-Order Cone Quadratic Eigenvalue Complementarity Problem (SOCQEiCP), which consists of finding a real number λ and a vector $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$w = \lambda^2 Ax + \lambda Bx + Cx \tag{1.16a}$$

$$x \in K,\ w \in K^* \tag{1.16b}$$

$$x^\top w = 0, \tag{1.16c}$$

where A, B, and $C \in \mathbb{R}^{n\times n}$ and where K is defined in (1.10)–(1.14). Similar to the SOCEiCP [12], in order to guarantee a nonzero complementary eigenvector, the following normalization constraint is added to the problem:

$$e^\top x - 1 = 0, \tag{1.17}$$

where $e = (e^1, e^2, \dots, e^r) \in \mathbb{R}^n$, $e^i = (1, 0, \dots, 0)^\top \in \mathbb{R}^{n_i}$ and $i = 1, \dots, r$. We design an appropriate enumerative method for finding a solution of SOCQEiCP, which computes a global minimum of an NLP formulation of a $2n$-dimensional SOCEiCP [5] that is equivalent to the SOCQEiCP. Such a formulation computes a positive eigenvalue for the SOCQEiCP, but the same approach can be applied for computing negative eigenvalues with a slight modification in the formulated $2n$-dimensional SOCEiCP. The choice of computing a positive eigenvalue is based on practical usefulness. For instance, for the mechanical contact problem presented in [27], the computation of positive eigenvalues predicts the presence of unstable modes in mechanical structures.

Although the enumerative algorithm is designed for a special SOCEiCP, it is different from the enumerative algorithm introduced for the SOCEiCP in [12]. First, the objective function of the NLP problem exploits the definition of the 2n-dimensional SOCEiCP and is therefore different from the one introduced in [12]. A crucial point for the success of the enumerative method is the use of the Reformulation-Linearization Technique (RLT) bound-factor constraints. Such constraints can be formulated once appropriate bounds for variables are computed. The procedures for the derivation of these bounds are based on certain sufficient conditions for the matrices A and C proposed in this paper, which are different from the ones discussed in [12]. The convergence of the enumerative algorithm is also different from the one established before for the enumerative method in [12] and is included in this paper. Similar to the SOCEiCP, a semi-smooth Newton method is developed for the SOCQEiCP that exploits the special structure of the 2n-dimensional SOCEiCP equivalent to the SOCQEiCP. Furthermore, a hybrid method that combines the enumerative method and the semi-smooth Newton method is also designed for enhancing the computational efficiency of using just the former (convergent) algorithm.

We note that the resolution of the eigenvalue problems EiCP and QEiCP is important in many models of dynamic analysis of structural mechanical systems, vibro-acoustic systems, electrical circuit simulation, fluid dynamics and contact problems in mechanics (see, for instance, [27, 31]). In some applications, the knowledge of eigenvalues can avoid the instability and unwanted resonance for a given system. In mechanical structures, for example, eigenvalues are related to resonance frequency and to the stability analysis of the corresponding dynamical systems. The computation of eigenvalues becomes crucial to identify damped eigenvalues corresponding to unstable modes or large vibrations. Recently, it was shown that a wide range of applications in engineering design, transportation science, game theory, and economic equilibrium, can be formulated as optimization problems involving second-order cone constraints [23]. Hence, we believe that the combination of eigenvalue complementarity problems and second-order cone programming problems may be used to model a large class of practical applications involving engineering design, stability analysis in game theory or equilibrium contexts. A detailed study of such applications is recommended for further investigation.

The remainder of this paper is organized as follows. In Section 2, we first recall the results established in [5], which reduce the SOCQEiCP into a $2n$-dimensional SOCEiCP under some sufficient conditions on the matrices A and C. As for the SOCEiCP, we introduce an NLP formulation for the $2n$-dimensional SOCEiCP in Section 3, and we establish a necessary and sufficient condition for an SP of this NLP to be a solution of SOCQEiCP. An enumerative algorithm is next proposed and analyzed in Sections 4 and 5 in order to provably solve the SOCQEiCP by computing a global minimum of the equivalent NLP formulation. The semi-smooth Newton method and the hybrid algorithm are discussed in Sections 6 and 7, respectively. Numerical results with a number of test problems are reported in Section 8 in order to illustrate the efficiency of the hybrid method in practice, and Section 9 closes the paper with some concluding remarks.

2 A $2n$-Dimensional SOCEiCP

Consider again the SOCQEiCP given by (1.16). Similar to [5], we impose the following (not too restrictive) conditions on the matrices A and C:

(A1) The matrix A is positive definite (PD), i.e.,

$$x^\top Ax > 0, \ \forall x \neq 0.$$

(A2) $C \in S_0'$ matrix, i.e., $x = 0$ is the unique feasible solution of

$$x \in K \tag{2.1a}$$

$$Cx \in K. \tag{2.1b}$$

Note that a matrix C is S_0' if there is no $x \neq 0$ satisfying the conditions (2.1) and the regularization constraint (1.17). Now, consider the following $2n$-dimensional SOCEiCP on $K \times K$ as defined in [5] along with the normalization constraint (1.17):

$$\lambda \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} - \begin{bmatrix} -B & -C \\ I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} w \\ t \end{bmatrix} \tag{2.2a}$$

$$y^\top w + x^\top t = 0 \tag{2.2b}$$

$$x, y, w, t \in K \tag{2.2c}$$

$$e^\top (x + y) = 1. \tag{2.2d}$$

Then the following property holds [5]:

Proposition 2.1. (i) *The SOCEiCP* (2.2) *has at least one solution* (λ, z)*, with* $z = (x, y) \in \mathbb{R}^{2n}$*.*

(ii) *In any solution of the SOCEiCP* (2.2)*,* $t = 0$ *and* $\lambda > 0$*.*

(iii) *If* (λ, z) *is a solution of the SOCEiCP* (2.2) *with* $z = (y, x) \in \mathbb{R}^{2n}$*, then* $(\lambda, (1 + \lambda)x)$ *solves SOCQEiCP* (1.16)*.*

As analyzed in [5], a negative eigenvalue for SOCQEiCP can be guaranteed if B replaces $-B$ in the definition of the $2n$-dimensional SOCEiCP.

Many optimization textbooks [3,26] discuss the importance of scaling in order to improve the numerical accuracy of the solutions computed by optimization algorithms. We define the following diagonal matrix:

$$D = \frac{1}{\alpha} I_n, \tag{2.3}$$

where I_n is the identity matrix of order n, and where

$$\alpha = \sqrt{\max\{|a_{ij}|, |b_{ij}|, |c_{ij}|\}}, \tag{2.4}$$

$i = 1, \dots, n$ and $j = 1, \dots, n$. Then the following properties can be easily shown.

Proposition 2.2. (i) *A is PD if and only if DAD is PD.*

(ii) *C is S_0' if and only DCD is S_0'.*

Due to Proposition 2.2, the SOCQEiCP(DAD, DBD, DCD) satisfies the assumptions $(A1)$ and $(A2)$ if $A \in$ PD and C belongs to S_0'. Therefore, one can always reduce the SOCQEiCP to one where the elements of the matrix of the problem belong to the interval $[-1, 1]$. It is easy to show that this scaled SOCQEiCP has the same eigenvalues of the original problem but the eigenvectors are scaled by a factor $1/\alpha$, where α is given by (2.4).

3 A Nonlinear Programming Formulation for SOCQEiCP

In this section, we propose an equivalent nonlinear programming formulation for the $2n$-dimensional SOCEiCP. By following the approach given in [4] and [12], we introduce the vectors:

$$y_j^i = \lambda x_j^i, \; j = 0, 1, \dots, n_i - 1, \; i = 1, \dots, r, \tag{3.1}$$

$$v_j^i = \lambda y_j^i,\ j = 0, 1, \dots, n_i - 1,\ i = 1, \dots, r, \tag{3.2}$$

where (3.1) follows from the second row in (2.2a), noting that $t = 0$. Since $\lambda > 0$ and $t = 0$ in any solution to SOCEiCP (2.2), then $y^\top w = x^\top w = v^\top w = 0$ in such a solution. This leads to the consideration of the following nonlinear program:

$$\begin{aligned}
\mathbf{NLP}_1 : \text{Minimize}\quad & f(x, y, v, w, \lambda) = \|y - \lambda x\|^2 + \|v - \lambda y\|^2 + (x^\top w)^2 + (y^\top w)^2 \\
& \qquad + (v^\top w)^2 && (3.3a) \\
\text{subject to}\quad & w = Av + By + Cx && (3.3b) \\
& \|\bar{x}^i\|^2 \le (x_0^i)^2, \quad i = 1, \dots, r && (3.3c) \\
& \|\bar{y}^i\|^2 \le (y_0^i)^2, \quad i = 1, \dots, r && (3.3d) \\
& \|\bar{v}^i\|^2 \le (v_0^i)^2, \quad i = 1, \dots, r && (3.3e) \\
& \|\bar{w}^i\|^2 \le (w_0^i)^2, \quad i = 1, \dots, r && (3.3f) \\
& e^\top (x + y) = 1 && (3.3g) \\
& e^\top (y + v) = \lambda && (3.3h) \\
& x_0^i \ge 0,\ i = 1, \dots, r && (3.3i) \\
& y_0^i \ge 0,\ i = 1, \dots, r && (3.3j) \\
& v_0^i \ge 0,\ i = 1, \dots, r && (3.3k) \\
& w_0^i \ge 0,\ i = 1, \dots, r && (3.3l)
\end{aligned}$$

where $w^i = (w_0^i, \bar{w}^i) \in \mathbb{R}^{n_i}$, $y^i = (y_0^i, \bar{y}^i), \in \mathbb{R}^{n_i}$, $v^i = (v_0^i, \bar{v}^i), \in \mathbb{R}^{n_i}$ for $i = 1, \dots, r$, and $w = (w^1, w^2, \dots, w^r) \in \mathbb{R}^n$, $y = (y^1, y^2, \dots, y^r) \in \mathbb{R}^n$, and $v = (v^1, v^2, \dots, v^r) \in \mathbb{R}^n$.

Remark 3.1. Whereas the nonlinear problem $\mathbf{NLP}_1$ can be rewritten using a cone-wise formalism, we prefer to use the component-wise notation adopted above, because it is in line with the enumerative method presented in Section 5.

Proposition 3.2. *The nonlinear problem $\boldsymbol{NLP}_1$ in* (3.3) *has a global minimum $(x^*, y^*, v^*, w^*, \lambda^*)$ such that $f(x^*, y^*, v^*, w^*, \lambda^*) = 0$ if and only if (λ^*, x^*, y^*) is a solution of SOCEiCP* (2.2) *with $\lambda^* > 0$ and $t^* = 0$.*

Proof. If the optimal value of $\mathbf{NLP}_1$ is equal to zero, all the constraints of the SOCEiCP (2.2) are satisfied with $t^* = 0$. Note that if $\lambda^* = 0$ then $y^* = v^* = 0$ by (3.3d), (3.3e), (3.3h), (3.3j), and (3.3k), and this contradicts the assumption (A2) by (3.3b), (3.3c), (3.3f), (3.3g), and (3.3i). On the other hand, since $y^* = \lambda^* x^*$, then $\lambda^* < 0$ is impossible by (3.3g), (3.3i), and (3.3j). The sufficiency implication is obvious. □

Since any global minimum of $\mathbf{NLP}_1$ is a stationary point (see Remark 3.4 below) and a stationary point is much easier to compute, it is interesting to investigate when a stationary point of $\mathbf{NLP}_1$ provides a solution of SOCQEiCP. The following proposition addresses this issue.

Proposition 3.3. *A given stationary point $(x^*, y^*, v^*, w^*, \lambda^*)$ of $\boldsymbol{NLP}_1$ (satisfying the KKT conditions) is a global minimum of the nonlinear problem $\boldsymbol{NLP}_1$* (3.3) *with $f(x^*, y^*, v^*, w^*, \lambda^*) = 0$ (i.e., a solution to SOCQEiCP) if and only if the Lagrange multipliers associated with the constraints* (3.3g) *and* (3.3h) *are equal to zero.*

Proof. Let $\alpha \in \mathbb{R}^n$, $\beta \in \mathbb{R}^r$, $\mu \in \mathbb{R}^r$, $\sigma \in \mathbb{R}^r$, $\zeta \in \mathbb{R}^r$, $\gamma \in \mathbb{R}$, $\xi \in \mathbb{R}$, $\delta \in \mathbb{R}^r$, $\theta \in \mathbb{R}^r$, $\nu \in \mathbb{R}^r$, and $\rho \in \mathbb{R}^r$ be the Lagrange multipliers associated with the constraints (3.3b)–(3.3l), respectively.

Define

$$D = \begin{bmatrix} 2x_0^1 & 0 & \cdots & 0 \\ -2\bar{x}^1 & 0 & \cdots & 0 \\ 0 & 2x_0^2 & \cdots & 0 \\ 0 & -2\bar{x}^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 2x_0^r \\ 0 & 0 & \cdots & -2\bar{x}^r \end{bmatrix} \in \mathbb{R}^{n\times r},\ E = \begin{bmatrix} e^1 & 0 & \cdots & 0 \\ 0 & e^2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & e^r \end{bmatrix} \in \mathbb{R}^{n\times r}, \tag{3.4a}$$

$$F = \begin{bmatrix} 2y_0^1 & 0 & \cdots & 0 \\ -2\bar{y}^1 & 0 & \cdots & 0 \\ 0 & 2y_0^2 & \cdots & 0 \\ 0 & -2\bar{y}^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 2y_0^r \\ 0 & 0 & \cdots & -2\bar{y}^r \end{bmatrix} \in \mathbb{R}^{n\times r},\ H = \begin{bmatrix} 2w_0^1 & 0 & \cdots & 0 \\ -2\bar{w}^1 & 0 & \cdots & 0 \\ 0 & 2w_0^2 & \cdots & 0 \\ 0 & -2\bar{w}^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 2w_0^r \\ 0 & 0 & \cdots & -2\bar{w}^r \end{bmatrix} \in \mathbb{R}^{n\times r}, \tag{3.4b}$$

$$L = \begin{bmatrix} 2v_0^1 & 0 & \cdots & 0 \\ -2\bar{v}^1 & 0 & \cdots & 0 \\ 0 & 2v_0^2 & \cdots & 0 \\ 0 & -2\bar{v}^2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 2v_0^r \\ 0 & 0 & \cdots & -2\bar{v}^r \end{bmatrix} \in \mathbb{R}^{n\times r}. \tag{3.4c}$$

A KKT (stationary) point (x, y, v, w, λ) of the problem $\mathbf{NLP}_1$ satisfies the following conditions [3, 26]:

$$-2\lambda(y - \lambda x) + 2(x^\top w)w = -C^\top \alpha + D\beta + E\delta + \gamma e \tag{3.5a}$$

$$2(y - \lambda x) - 2\lambda(v - \lambda y) + 2(y^\top w)w = -B^\top \alpha + F\mu + E\theta + \gamma e + \xi e \tag{3.5b}$$

$$2(v - \lambda y) + 2(v^\top w)w = -A^\top \alpha + \xi e + L\gamma + E\nu \tag{3.5c}$$

$$2(x^\top w)x + 2(y^\top w)y + 2(v^\top w)v = \alpha + H\zeta + E\rho \tag{3.5d}$$

$$-2x^\top(y - \lambda x) - 2y^\top(v - \lambda y) = -\xi \tag{3.5e}$$

$$\beta_i[\|\bar{x}_i\|^2 - (x_0^i)^2] = 0, \quad i = 1, \ldots, r \tag{3.5f}$$

$$\mu_i[\|\bar{y}_i\|^2 - (y_0^i)^2] = 0, \quad i = 1, \ldots, r \tag{3.5g}$$

$$\sigma_i[\|\bar{v}_i\|^2 - (v_0^i)^2] = 0, \quad i = 1, \ldots, r \tag{3.5h}$$

$$\zeta_i[\|\bar{w}_i\|^2 - (w_0^i)^2] = 0, \quad i = 1, \ldots, r \tag{3.5i}$$

$$\delta_i x_0^i = \theta_i y_0^i = \nu_i v_0^i = \rho_i w_0^i = 0, \quad i = 1, \ldots, r \tag{3.5j}$$

$$\beta_i \geq 0,\ \mu_i \geq 0,\ \sigma_i \geq 0,\ \zeta_i \geq 0,\ \delta_i \geq 0,\ \theta_i \geq 0,\ \nu_i \geq 0,\ \rho_i \geq 0, \quad i = 1, \ldots, r, \tag{3.5k}$$

where β_i, μ_i, σ_i, ζ_i, δ_i, θ_i, ν_i, and ρ_i are the i-th components of the vectors β, μ, σ, ζ, δ, θ,

ν, and $\rho \in \mathbb{R}^r$, respectively. By multiplying both sides of (3.5a), (3.5b), (3.5c), and (3.5d) by $x^\top$, $y^\top$, $v^\top$, and $w^\top$, respectively, and by using (3.5j), we have

$$\begin{aligned}
&-2\lambda x^\top(y-\lambda x) + 2(x^\top w)^2 = -\alpha^\top Cx + 2\sum_{i=1}^r \beta_i(-\|\bar{x}_i\|^2 + (x_0^i)^2) + \gamma x^\top e \\
&2y^\top(y-\lambda x) - 2\lambda y^\top(v-\lambda y) + 2(y^\top w)^2 = -\alpha^\top By + 2\sum_{i=1}^r \mu_i(-\|\bar{y}_i\|^2 + (y_0^i)^2) + \gamma y^\top e \\
&\qquad\qquad\qquad\qquad + \xi y^\top e \\
&2v^\top(v-\lambda y) + 2(v^\top w)^2 = -\alpha^\top Av + \xi v^\top e + 2\sum_{i=1}^r \sigma_i(-\|\bar{v}_i\|^2 + (v_0^i)^2) \\
&2(x^\top w)^2 + 2(y^\top w)^2 + 2(v^\top w)^2 = \alpha^\top w + \sum_{i=1}^r \zeta_i(-\|\bar{w}_i\|^2 + (w_0^i)^2).
\end{aligned}$$

By adding the above equalities and by using (3.3b), (3.3g), (3.3h), (3.5f), (3.5g), (3.5h), and (3.5i), we get

$$2(x^\top w)^2 + 2(y^\top w)^2 + 2(v^\top w)^2 + 2f(x,y,v,w,\lambda) = \gamma + \xi\lambda. \tag{3.7}$$

If $\gamma = 0$ and $\xi = 0$, then the objective function value is zero, which means that the stationary point is a solution of SOCQEiCP. Conversely, suppose that (x,y,w,λ) is a solution of SOCQEiCP. Then, by Proposition 3.2, $f(x,y,v,w,\lambda)$ is null and the same holds for the terms $(x^\top w)^2$, $(y^\top w)^2$, and $(v^\top w)^2$. Since $f(x,y,v,w,\lambda) = 0$, we have $y = \lambda x$ and $v = \lambda y$, and so $\xi = 0$ from (3.5e) and $\gamma = 0$ from (3.7). □

Remark 3.4. Note that Proposition 3.3 addresses a given stationary point for $\mathbf{NLP}_1$ satisfying the KKT conditions. However, it is important to note that for any given solution (x^*, w^*, λ^*) to SOCQEiCP, there corresponds a KKT (stationary) point $(x^*, y^*, v^*, w^*, \lambda^*)$ of $\mathbf{NLP}_1$. This follows without the need for verifying any constraint qualification, since $(x^*, y^*, v^*, w^*, \lambda^*)$ with $y^* = \lambda^* x^*$ and $v^* = \lambda^* y$ is a feasible solution to $\mathbf{NLP}_1$ at which the gradient of the objective function vanishes, and hence a KKT point.

4 Additional Constraints for the Nonlinear Programming Formulation

Following the approach in [12], we show how to compute compact intervals for the variables involved in the enumerative algorithm to be described in Section 5. In particular, we impose the following bounds on the variables:

$$c \le x \le d \tag{4.1a}$$

$$g \le y \le h \tag{4.1b}$$

$$l \le \lambda \le u \tag{4.1c}$$

$$L \le w \le U, \tag{4.1d}$$

where $c = [c_j^i]$, $d = [d_j^i]$, $g = [g_j^i]$, $h = [h_j^i]$, $L = [L_j^i]$, and $U = [U_j^i]$, $j = 0, 1, \ldots, n_i - 1$, $i = 1, \ldots, r$. In what follows, we show how to compute the foregoing bounds, and we embed these within an enumerative search process based on the Reformulation-Linearization Technique [30].

$\boxed{4.1}$ Lower and upper bounds for the x- and y-variables

Any feasible vectors x and y in the formulation $\mathbf{NLP}_1$ belong to the set

$$\begin{aligned}\Delta = \{(x,y) \in \mathbb{R}^{2n} : e^\top (x+y) = 1,\ x_0^i \geq 0,\ -1 \leq x_j^i \leq 1,\ y_0^i \geq 0,\\ -1 \leq y_j^i \leq 1,\ j = 1, \ldots, n_i - 1,\ i = 1, \ldots, r\}.\end{aligned} \tag{4.2}$$

Accordingly, lower and upper bounds for the variables x and y can be set as

$$g_0^i = c_0^i = 0,\ h_0^i = d_0^i = 1, \quad i = 1, \ldots, r \tag{4.3a}$$

$$g_j^i = c_j^i = -1,\ h_j^i = d_j^i = 1, \quad j = 1, \ldots, n_i - 1,\ i = 1, \ldots, r. \tag{4.3b}$$

$\boxed{4.2}$ Upper bound for the variable λ

The next result provides an upper bound for the complementarity eigenvalue λ.

Theorem 4.1. *Let* $\mu = \sum_{i=1}^n \left(\sum_{j=i}^n |b_{ij}| + |c_{ij}| \right) + 1$. *Then we can take*

$$u = \frac{\mu}{\bar{y}^\top A \bar{y}^\top + \bar{x}^\top \bar{x}}, \tag{4.4}$$

where $(\bar{x}, \bar{y})$ *is a global minimum of the following problem*

$$\begin{aligned} &\textit{Minimize} \quad y^\top A y + x^\top x \\ &\textit{subject to} \quad (x,y) \in \Delta, \end{aligned} \tag{4.5}$$

where Δ *is given by* (4.2).

Proof. See [12] for the proof. □

Due to the assumption (A1), the problem (4.5) is a strictly convex quadratic problem. Hence, this program has a unique optimal solution, which is a stationary point of the objective function in the simplex Δ.

$\boxed{4.3}$ Lower bound for the variable λ

Consider the following convex nonlinear program:

$$\mathbf{NLP}_2 : \quad \text{Minimize} \quad \sum_{i=1}^r (y_0^i + v_0^i) \tag{4.6a}$$

$$\text{subject to } w = Av + By + Cx \tag{4.6b}$$

$$(x,y) \in \Delta \tag{4.6c}$$

$$L_0^i \leq w_0^i \leq U_0^i,\ i = 1, \ldots, r \tag{4.6d}$$

$$v_0^i \geq 0,\ i = 1, \ldots, r \tag{4.6e}$$

$$\|\bar{x}^i\|^2 \leq (x_0^i)^2,\ i = 1, \ldots, r \tag{4.6f}$$

$$\|\bar{y}^i\|^2 \leq (y_0^i)^2,\ i = 1, \ldots, r \tag{4.6g}$$

$$\|\bar{v}^i\|^2 \leq (v_0^i)^2,\ i = 1, \ldots, r \tag{4.6h}$$

$$\|\bar{w}^i\|^2 \leq (w_0^i)^2,\ i = 1, \ldots, r, \tag{4.6i}$$

where L_0^i and U_0^i are, respectively, some finite lower and upper bounds for the variable w_0^i, which are derived in Section 4.4.

An optimal solution to $\mathbf{NLP}_2$ provides the required lower bound l for the variable λ. Note that $\mathbf{NLP}_2$ is convex (noting that (4.6f)–(4.6i) are equivalent to the corresponding convex Lorentz cone constraints), which means that a stationary (KKT) point gives a global minimum. This fact is a consequence of Propositions 4.2 and 4.3 stated below.

Proposition 4.2. *$\boldsymbol{NLP}_2$ has an optimal solution.*

Proof. Let $(\tilde{x},\ \tilde{y}) \in \Delta$ satisfying (4.6f) and (4.6g) and let $\tilde{w}$ satisfying (4.6d) and (4.6i). Hence, $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{v})$ is a feasible solution of $\mathbf{NLP}_2$, where $\tilde{v}$ is the unique solution of the linear system $A\tilde{v} = \tilde{w} - B\tilde{y} - C\tilde{x}$ ($A \in$ PD). So it remains to show that $\mathbf{NLP}_2$ has no nonzero recession direction $d = [d_x, d_y, d_w, d_v]^\top$, where d_x, d_y, d_w, d_v are the components of d corresponding to the x-, y-, w- and v-variables, respectively. From (4.6c), (4.6d) and (4.6i), any such recession direction must satisfy $d_x = d_y = d_w = 0$ and from (4.6b) we have $Ad_v = 0$, which yields $d_v = 0$ because $A \in$ PD. Thus the feasible region of $\mathbf{NLP}_2$ is nonempty and bounded, and so $\mathbf{NLP}_2$ has an optimal solution. □

Proposition 4.3. *If $C \in S_0'$, then $\boldsymbol{NLP}_2$ has a positive optimal value.*

Proof. $\mathbf{NLP}_2$ has a zero optimal value if and only if $y_0^i = v_0^i = 0$ for all $i = 1, \ldots, r$, which implies together with (4.6g) and (4.6h) that $v = y = 0$. Hence there must exist vectors w and x, such that $w = Cx$ and the constraints (4.6b), (4.6c), (4.6f), (4.6i) hold. This is impossible, because of assumption (A2). Thus, if $C \in S_0'$, we conclude that the lower bound l is strictly positive. □

4.4 Lower and upper bounds for the w-variables

In this section, we compute the bounds for each of the r sets of variables w_0^i and for $\bar{w}^i$. First of all, $w_0^i \geq 0 \equiv L_0^i$ for $i = 1, \ldots, r$. Moreover, from the equation

$$w = \lambda^2 Ax + \lambda Bx + Cx, \tag{4.7}$$

we have

$$w_0^i = \sum_{j=1}^{n} (\lambda^2 a_{t_i,j} + \lambda b_{t_i,j} + c_{t_i,j}) x_j, \quad i = 1, \ldots, r, \tag{4.8}$$

where $t_1 = 1$ and $t_i = 1 + \sum_{k=1}^{i-1} n_k$, $i = 2, \ldots, r$. Hence, by (4.3)

$$w_0^i \leq \sum_{j=1}^{n} (u^2 |a_{t_i,j}| + u |b_{t_i,j}| + |c_{t_i,j}|) \equiv U_0^i, \quad i = 1, \ldots, r. \tag{4.9}$$

Since

$$\|\bar{w}^i\| \leq w_0^i,\ i = 1, \ldots, r, \tag{4.10}$$

we get the following lower and upper bounds for the variables w_j^i:

$$L_j^i \equiv -U_0^i \leq w_j^i \leq U_0^i \equiv U_j^i,\ j = 1, \ldots, n_i - 1,\ i = 1, \ldots, r. \tag{4.11}$$

Note that L_j^i and U_j^i, $j = 0, \ldots, n_i$, $i = 1, \ldots, r$ depend on u, which is the upper bound of the variable λ. Such a value could be modified during the performance of the enumerative method. Therefore, at each node of the enumerative method, the bounds for the w-variables are updated by using the current value of u at that node.

$\boxed{4.5}$ Reformulation-Linearization Technique (RLT) constraints

Given the lower and the upper bounds in (4.1), we can incorporate additional RLT-based constraints [30] within the nonlinear problem $\mathbf{NLP}_1$ in order to design the enumerative method presented in the next section. We begin by introducing the following n additional variables:

$$z_j^i \equiv x_j^i w_j^i,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r. \tag{4.12}$$

By using the approach in [30], we define nonnegative bound-factors for the x-, y-, w-, and λ-variables as follows: $(x - c)$ and $(d - x)$; $(y - g)$ and $(h - y)$; $(w - L)$ and $(U - w)$; and, $(\lambda - l)$ and $(u - \lambda)$. Then we generate the so-called bound-factor RLT constraints by considering the following product restrictions:

$$[c_j^i \le x_j^i \le d_j^i] * [L_j^i \le w_j^i \le U_j^i],\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.13}$$

$$[c_j^i \le x_j^i \le d_j^i] * [l \le \lambda \le u],\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.14}$$

$$[g_j^i \le y_j^i \le h_j^i] * [l \le \lambda \le u],\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r. \tag{4.15}$$

In (4.13), we consider the nonnegative product of each of the two bound-factors associated with the x_j^i-variable (i.e., $(x_j^i - c_j^i)$ and $(d_j^i - x_j^i)$) with each of the two bound-factors associated with the w_j^i-variable (i.e., $(w_j^i - L_j^i)$ and $(U_j^i - w_j^i)$), for each $j = 0, 1, \ldots, n_i - 1$, $i = 1, \ldots, r$, which are subsequently linearized using the substitutions specified in (4.12). In the same way, we consider the nonnegative products of the bound-factors associated with the x-variables (i.e., $(x_j^i - c_j^i)$ and $(d_j^i - x_j^i)$) and y-variables (i.e., $(y_j^i - g_j^i)$ and $(h_j^i - y_j^i)$) with the bound-factors for the λ-variable (i.e., $(\lambda - l)$ and $(u - \lambda)$) together with the substitutions (3.1) and (3.2). The following resulting $12n$ constraints are then incorporated within the nonlinear program $\mathbf{NLP}_1$:

$$z_j^i \ge c_j^i w_j^i + L_j^i x_j^i - c_j^i L_j^i,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16a}$$

$$z_j^i \ge d_j^i w_j^i + U_j^i x_j^i - d_j^i U_j^i,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16b}$$

$$z_j^i \le c_j^i w_j^i + U_j^i x_j^i - c_j^i U_j^i,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16c}$$

$$z_j^i \le d_j^i w_j^i + L_j^i x_j^i - d_j^i U_j^i,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16d}$$

$$y_j^i \ge x_j^i l + c_j^i \lambda - c_j^i l,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16e}$$

$$y_j^i \ge x_j^i u + d_j^i \lambda - d_j^i u,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16f}$$

$$y_j^i \le x_j^i u + c_j^i \lambda - c_j^i u,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16g}$$

$$y_j^i \le x_j^i l + d_j^i \lambda - d_j^i l,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16h}$$

$$v_j^i \ge y_j^i l + g_j^i \lambda - g_j^i l,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16i}$$

$$v_j^i \ge y_j^i u + h_j^i \lambda - h_j^i u,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16j}$$

$$v_j^i \le y_j^i u + g_j^i \lambda - g_j^i u,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r \tag{4.16k}$$

$$v_j^i \le y_j^i l + h_j^i \lambda - h_j^i l,\ j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r. \tag{4.16l}$$

The complementarity constraint $x^\top w = \sum_{i=1}^r (x^i)^\top w^i$ is the sum of nonnegative terms, noting that

$$(x^i)^\top w^i = x_0^i w_0^i + (\bar{x}^i)^\top \bar{w}^i \ge x_0^i w_0^i - \|\bar{x}^i\| \|\bar{w}^i\| \ge 0,\ i = 1, \ldots, r. \tag{4.17}$$

This means that to have $x^\top w = 0$ with $x \in K$ and $w \in K$, we must have $(x^i)^\top w^i = 0$ for $i = 1, \dots, r$. So we can remove the quadratic term $(x^\top w)^2$ from the objective function and, instead, add the term shown in (4.19a) along with the following r linear constraints:

$$\sum_{j=0}^{n_i-1} z_j^i = 0, \ i = 1, \dots, r. \tag{4.18}$$

Accordingly, the nonlinear programming formulation of SOCQEiCP, which we propose to solve by means of the enumerative method presented in the next section, is given as follows:

$$\begin{array}{lll}
\mathbf{NLP}_3 : \text{Minimize} & \widetilde{f}(x, y, v, w, \lambda, z) = \|y - \lambda x\|^2 + \|v - \lambda y\|^2 + \|z - x \circ w\|^2 & \\
& \qquad\qquad + (y^\top w)^2 + (v^\top w)^2 & (4.19\text{a}) \\
\text{s.t.} & w = Av + By + Cx & (4.19\text{b}) \\
& \|\bar{x}^i\|^2 \le (x_0^i)^2, \quad i = 1, \dots, r & (4.19\text{c}) \\
& \|\bar{y}^i\|^2 \le (y_0^i)^2, \quad i = 1, \dots, r & (4.19\text{d}) \\
& \|\bar{v}^i\|^2 \le (v_0^i)^2, \quad i = 1, \dots, r & (4.19\text{e}) \\
& \|\bar{w}^i\|^2 \le (w_0^i)^2, \quad i = 1, \dots, r & (4.19\text{f}) \\
& e^\top (x + y) = 1 & (4.19\text{g}) \\
& e^\top (y + v) = \lambda & (4.19\text{h}) \\
& (4.1) & (4.19\text{i}) \\
& (4.16) & (4.19\text{j}) \\
& (4.18) & (4.19\text{k})
\end{array}$$

where $\circ$ is the Hadamard product. Note that $\mathbf{NLP}_3$ is a convex constrained program with a nonconvex objective function, where (4.19c)–(4.19f) are equivalent to the corresponding Lorentz cone inclusion constraints.

Similar to Proposition 3.2 for the nonlinear problem $\mathbf{NLP}_1$, the following results hold for $\mathbf{NLP}_3$:

Proposition 4.4. *SOCQEiCP has a solution $(\widetilde{x}, \widetilde{w}, \widetilde{\lambda})$ if and only if $(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ is a global minimum of $\boldsymbol{NLP}_3$ with $\widetilde{f}(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z}) = 0$.*

Proposition 4.5. *For any given solution (x^*, w^*, λ^*) to SOCQEiCP, there corresponds a stationary point $(x^*, y^*, v^*, w^*, \lambda^*, z^*)$ of $\boldsymbol{NLP}_3$.*

5 An Enumerative Method

In this section, we introduce an enumerative algorithm for finding a global minimum to the nonlinear problem $\mathbf{NLP}_3$. This is done by exploring a binary tree that is constructed by partitioning the intervals $[c_j^i, d_j^i]$ associated with the variables x_j^i, $j = 0, 1, \dots, n_i - 1$, $i = 1, \dots, r$ and the interval $[l, u]$ associated with the variable λ. The steps of the enumerative method are as follows:

Algorithm 1 Enumerative algorithm for SOCQEiCP

▷ **Step** 0 **(Initialization)**
Set $\epsilon_1 > 0$ and $\epsilon_2 > 0$. Set $k = 1$ and find a stationary point $(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ of $\mathbf{NLP}_3(1)$. If $\mathbf{NLP}_3(1)$ is infeasible, then SOCQEiCP has no solution; terminate. Otherwise, let $P = \{1\}$ be the set of open nodes, set $UB(1) = f(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ and let $N = 1$ be the number of generated nodes.

▷ **Step** 1 **(Choice of node)**
If $P = \emptyset$ terminate; SOCQEiCP has no solution. Otherwise, select $k \in P$ such that $UB(k) = \min\{UB(i) : i \in P\}$, set $P = P \setminus \{k\}$, and let $(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ be the stationary point that was previously found at this node.

▷ **Step** 2 **(Branching rule)**
Let

$$\theta_1 = \max\left\{|\widetilde{z}^i_j - \widetilde{x}^i_j \widetilde{w}^i_j| : j = 0, 1, \ldots, n_i - 1, i = 1, \ldots, r\right\} \tag{5.1}$$

$$\theta_2 = \max\left\{|\widetilde{y}^i_j - \widetilde{\lambda}\widetilde{x}^i_j|, |\widetilde{v}^i_j - \widetilde{\lambda}\widetilde{y}^i_j| : j = 0, 1, \ldots, n_i - 1,\ i = 1, \ldots, r\right\} \tag{5.2}$$

and let the maximum in (5.1) be achieved by (i^*, j^*).

(i) If $\theta_1 \leq \epsilon_1$ and $\theta_2 \leq \epsilon_2$ then $\widetilde{\lambda}$ yields a quadratic complementary eigenvalue within the tolerance ϵ_2 with $(1 + \widetilde{\lambda})\widetilde{x}$ being a corresponding quadratic eigenvector.

(ii) If $\theta_1 > \theta_2$ then partition the interval $[\widetilde{c}^{i^*}_{j^*}, \widetilde{d}^{i^*}_{j^*}]$ for the variable $\widetilde{x}^{i^*}_{j^*}$ at node k into $[\widetilde{c}^{i^*}_{j^*}, \hat{x}^{i^*}_{j^*}]$ and $[\hat{x}^{i^*}_{j^*}, \widetilde{d}^{i^*}_{j^*}]$ to generate two new nodes $N + 1$ and $N + 2$, where

$$\hat{x}^{i^*}_{j^*} = \begin{cases} \widetilde{x}^{i^*}_{j^*} & \text{if } \min\{(\widetilde{x}^{i^*}_{j^*} - \widetilde{c}^{i^*}_{j^*}), (\widetilde{d}^{i^*}_{j^*} - \widetilde{x}^{i^*}_{j^*})\} \geq 0.1(\widetilde{d}^{i^*}_{j^*} - \widetilde{c}^{i^*}_{j^*}) \\ \frac{\widetilde{d}^{i^*}_{j^*} + \widetilde{c}^{i^*}_{j^*}}{2} & \text{otherwise.} \end{cases} \tag{5.3}$$

(iii) If $\theta_1 \leq \theta_2$ then partition the interval $[\widetilde{l}, \widetilde{u}]$ for $\widetilde{\lambda}$ at node k into $[\widetilde{l}, \hat{\lambda}]$ and $[\hat{\lambda}, \widetilde{u}]$ to generate two new nodes $N + 1$ and $N + 2$, where

$$\hat{\lambda} = \begin{cases} \widetilde{\lambda} & \text{if } \min\{(\widetilde{\lambda} - \widetilde{l}), (\widetilde{u} - \widetilde{\lambda})\} \geq 0.1(\widetilde{u} - \widetilde{l}) \\ \frac{\widetilde{u} + \widetilde{l}}{2} & \text{otherwise.} \end{cases} \tag{5.4}$$

▷ **Step** 3 **(Solve, Update and Queue)**
For each of $\nu = N+1$ and $\nu = N+2$, find a stationary point $(\hat{x}, \hat{y}, \hat{v}, \hat{w}, \hat{\lambda}, \hat{z})$ of $\mathbf{NLP}_3(\nu)$. If $\mathbf{NLP}_3(\nu)$ is feasible, set $P = P \cup \{\nu\}$ and $UB(\nu) = f(\hat{x}, \hat{y}, \hat{v}, \hat{w}, \hat{\lambda}, \hat{z})$. Return to Step 1.

Below, we state the main convergence theorem for the foregoing enumerative algorithm for solving SOCQEiCP. The proof closely follows that in [21], but we include the details for the sake of insights and completeness.

Theorem 5.1. *The enumerative algorithm for* $\boldsymbol{NLP}_3$ *run with* $\epsilon_1 = 0$ *and* $\epsilon_2 = 0$ *either terminates finitely with a solution to SOCQEiCP, or else, an infinite branch-and-bound (B&B) tree is generated such that along any infinite branch of this tree, any accumulation point of the stationary points obtained for* $\boldsymbol{NLP}_3$ *solves SOCQEiCP.*

Proof. The case of finite termination is obvious. Hence, suppose that an infinite B&B tree is generated, and consider any infinite branch. For notational convenience, denote $\zeta \equiv (x, y, v, w, \lambda, z)$ and let $\{\zeta^s\}_S$, with $s \in S$, be a sequence of stationary points of $\mathbf{NLP}_3$ that correspond to nodes on this infinite branch. Then, by taking a subsequence if necessary, we may assume

$$\{\zeta^s\}_S \to \zeta^*, \ \{[c^s, d^s]\}_S \to [c^*, d^*], \text{ and } \{[l^s, u^s]\}_S \to [l^*, u^*],$$

where $[c^s, d^s]$ and $[l^s, u^s]$ respectively denote the vectors of bounds on x and λ at node $s \in S$ of the B&B tree. We will show that ζ^* yields a solution to SOCQEiCP.

Note that along the infinite branch under consideration, we either branch on λ infinitely often, or else, there exists some index-pair $(\hat{i}, \hat{j})$ such that we branch on the interval for $x_{\hat{j}}^{\hat{i}}$ infinitely often. Let us assume the latter (the case of branching on λ infinitely often is similar, as discussed below), and suppose that this sequence of partitions corresponds to nodes indexed by $s \in S_1 \subseteq S$. By the partitioning rule (5.3), since the interval length for $x_{\hat{j}}^{\hat{i}}$ decreases by a geometric ratio of at most 0.9 over $s \in S_1$, we have in the limit that

$$c_{\hat{j}}^{*\hat{i}} = d_{\hat{j}}^{*\hat{i}} = x_{\hat{j}}^{*\hat{i}} = \nu^*, \text{ say.} \tag{5.5}$$

Furthermore, from (5.5) and the RLT bound-factor constraints (4.16a)–(4.16d), we have in the limit that

$$z_{\hat{j}}^{*\hat{i}} = w_{\hat{j}}^{*\hat{i}} \nu^* = w_{\hat{j}}^{*\hat{i}} x_{\hat{j}}^{*\hat{i}}. \tag{5.6}$$

Moreover, by the selection of the index-pair $(\hat{i},\hat{j})$ for $s \in S_1$, via (5.1) and (5.2) and the branching selection rule, we get that $\theta_1 \to 0$ and so $\theta_2 \to 0$ as well. (The case of branching on λ infinitely often likewise leads to $l^* = u^*$ in the limit, which from (4.16e)–(4.16l) yields that (3.1) and (3.2) hold true in the limit, and so again both θ_1 and θ_2 approach zero in the limit.) Thus in either case, we get in the limit as $s \to \infty$, $s \in S_1$, that

$$z_j^{*i} = w_j^{*i} x_j^{*i}, \ y_j^{*i} = \lambda^* x_j^{*i}, \text{ and } v_j^{*i} = \lambda^* y_j^{*i}, \ j = 0, 1, \ldots, n_i - 1, \ i = 1, \ldots, r, \tag{5.7}$$

or that (3.1), (3.2), and (4.12) hold true in the limit at ζ^*. Consequently, the set of constraints (4.19b) yields from (5.7) that, in the limit, $w^* - A\lambda^* y^* - By^* - Cx^* = 0$, i.e., by applying the second set of identities in (5.7), we have

$$w^* = \lambda^{*2} Ax^* + \lambda^* Bx^* + Cx^*. \tag{5.8}$$

Furthermore, by (4.18) and (5.7), we get

$$x^{*\top} w^* = 0. \tag{5.9}$$

Likewise, from (4.19c)–(4.19f), (3.3i)–(3.3l), and (5.7), we get

$$x^* \in K \text{ and } w^* \in K. \tag{5.10}$$

Thus, (5.8)–(5.10) imply that the (x^*, w^*, λ^*)-part of ζ^* represents a solution to SOCQEiCP. □

There are a couple of insightful points worth noting in regard to the proof of Theorem 5.1. First, observe that by (5.7) and (5.9), we get that $f(\zeta^*) = 0$ in the limit, as expected by Proposition 4.4. Second, observe that for (5.7) to hold true, i.e., for (5.6) to be a consequence of (5.5) (and similarly for the case of branching infinitely often on λ variable), we need just one pair of the four constraints from (4.16a)–(4.16d) (and likewise, one pair from each of (4.16e)–(4.16h) and (4.16i)–(4.16l)). However, we carry the entire set (4.16) because they assert additional valid inequalities that serve to assist in the convergence process.

$\boxed{6}$ A Semi-Smooth Algorithm

In this section, we use a semi-smooth algorithm for solving the SOCQEiCP (2.2). Due to Proposition 3.2, we know that $t = 0$ and the complementarity constraints (2.2b) can be replaced by

$$(x^i)^\top t^i = (y^i)^\top w^i = 0, \quad i = 1, \dots, r. \tag{6.1}$$

As in [12], we use the so-called natural residual function $\varphi^i_{\text{NR}} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ associated with the second-order cone K_i, which is defined by

$$\varphi^i_{\text{NR}}(x^i, t^i) = x^i - P_{K_i}(x^i - t^i) \tag{6.2}$$

$$\varphi^i_{\text{NR}}(y^i, w^i) = y^i - P_{K_i}(y^i - w^i), \tag{6.3}$$

where $P_{K_i}(\eta^i)$ is the projection of a vector $\eta^i = (\eta^i_0, \bar{\eta}^i) \in \mathbb{R} \times \mathbb{R}^{n_i - 1}$ onto the second-order cone K_i for each $i = 1, \dots, r$, i.e.,

$$P_{K_i}(\eta^i) = \arg \min_{\tau^i \in K_i} \|\tau^i - \eta^i\|. \tag{6.4}$$

The natural residual function φ^i_{NR} satisfies the following relations:

$$\varphi^i_{\text{NR}}(x^i, t^i) = 0 \Leftrightarrow x^i \in K_i, t^i \in K_i, (x^i)^\top t^i = 0 \tag{6.5}$$

$$\varphi^i_{\text{NR}}(y^i, w^i) = 0 \Leftrightarrow y^i \in K_i, w^i \in K_i, (y^i)^\top w^i = 0. \tag{6.6}$$

Consider the functions $\Phi_1(x, t) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $\Phi_2(y, w) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\Phi_1(x, t) = \begin{bmatrix} \varphi^1_{\text{NR}}(x^1, t^1) \\ \vdots \\ \varphi^r_{\text{NR}}(x^r, t^r) \end{bmatrix} \text{ and } \Phi_2(y, w) = \begin{bmatrix} \varphi^1_{\text{NR}}(y^1, w^1) \\ \vdots \\ \varphi^r_{\text{NR}}(y^r, w^r) \end{bmatrix}. \tag{6.7}$$

Then the SOCQEiCP (2.2) can be reformulated as follows:

$$\Psi(x, y, w, t, \lambda) = \begin{bmatrix} \Phi_1(x, t) \\ \Phi_2(y, w) \\ (\lambda A + B)y + Cx - w \\ \lambda x - y - t \\ e^\top (x + y) - 1 \end{bmatrix} = 0. \tag{6.8}$$

Algorithm 2 given below describes the steps of the semi-smooth algorithm for finding a solution of (6.8). Here, the Clarke generalized Jacobian of Φ at (x, y, w, t, λ) has the following form:

$$GJ(x, y, w, t, \lambda) = \begin{bmatrix} I_n - \tilde{V} & 0 & 0 & \tilde{V} & 0 \\ 0 & I_n - \hat{V} & \hat{V} & 0 & 0 \\ C & (\lambda A + B) & -I_n & 0 & Ay \\ \lambda I_n & -I_n & 0 & -I_n & x \\ e^\top & e^\top & 0 & 0 & 0 \end{bmatrix}, \tag{6.9}$$

where I_n denotes the $n \times n$ identity matrix and $\tilde{V}, \hat{V} \in R^{n \times n}$ are given as follows:

$$\tilde{V} = \begin{bmatrix} \tilde{V}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \tilde{V}^r \end{bmatrix}, \quad \hat{V} = \begin{bmatrix} \hat{V}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \hat{V}^r \end{bmatrix}. \tag{6.10}$$

The matrices $\tilde{V}^i$ and $\hat{V}^i$, $i = 1, \dots, r$ can be explicitly computed as in [12], by considering the spectral factorization of vectors in $\mathbb{R}^n$ specified by the Jordan algebra [11]. The matrices $\tilde{V}^i$ and $\hat{V}^i$ may be set-valued, and so in our numerical experiments we chose one value in the set and used it in the computation of the Clarke generalized Jacobian.

Algorithm 2 Semi-smooth Newton algorithm

- ▷ **Step** 0 **(Initialization)**
  Let $(\hat{x}, \hat{y}, \hat{w}, \hat{t}, \hat{\lambda})$ be an initial point such that $(\hat{x}, \hat{y}) \in \Delta$ and let $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$ be selected positive tolerances.

- ▷ **Step** 1 **(Newton direction)**
  Compute the Clarke generalized Jacobian GJ at $(\hat{x}, \hat{y}, \hat{w}, \hat{t}, \hat{\lambda})$ as in (6.9). If $GJ(\hat{x}, \hat{y}, \hat{w}, \hat{t}, \hat{\lambda})$ is singular, stop, and terminate with an unsuccessful termination. Otherwise, find the semi-smooth Newton direction $[d_x \; d_y \; d_w \; d_t \; d_\lambda]^\top$ by solving

$$GJ(\hat{x}, \hat{y}, \hat{w}, \hat{t}, \hat{\lambda}) \begin{bmatrix} d_x \\ d_y \\ d_w \\ d_t \\ d_\lambda \end{bmatrix} = -\Psi(\hat{x}, \hat{y}, \hat{w}, \hat{t}, \hat{\lambda})$$

  with Ψ given in (6.8).

- ▷ **Step** 2 **(Update)**
  Compute the new point

$$\widetilde{x} = \hat{x} + d_x, \; \widetilde{y} = \hat{y} + d_y, \; \widetilde{w} = \hat{w} + d_w, \; \widetilde{t} = \hat{t} + d_t, \text{ and } \widetilde{\lambda} = \hat{\lambda} + d_\lambda$$

  and let $\hat{x} = \widetilde{x}$, $\hat{y} = \widetilde{y}$, $\hat{w} = \widetilde{w}$, $\hat{t} = \widetilde{t}$, and $\hat{\lambda} = \widetilde{\lambda}$. Compute $\Phi_1(\hat{x}, \hat{t})$ and $\Phi_2(\hat{y}, \hat{w})$ as in (6.7). If the conditions

$$\max\{\|\hat{w} - (\hat{\lambda}A + B)\hat{y} - C\hat{x}\|, \|\hat{t} - \hat{\lambda}\hat{x} + \hat{y}\|\} \leq \tilde{\epsilon}_1$$

  and

$$\max\{\|\Phi_1(\hat{x}, \hat{t})\|, \|\Phi_2(\hat{y}, \hat{w})\|\} \leq \tilde{\epsilon}_2$$

  hold, then stop with $\hat{\lambda}$ being a quadratic complementary eigenvalue, $\hat{t} = 0$ in this solution and $(1 + \hat{\lambda})\hat{x}$ being the corresponding quadratic complementary eigenvector (see Proposition 2.1*(iii)*). Otherwise, go to Step 1.

7 A Hybrid Method

In order to combine the benefits of the enumerative method (Algorithm 1) with that of the semi-smooth Newton method (Algorithm 2), (as borne by our computational results reported in Section 8), we also explore the following hybrid algorithm:

Algorithm 3 Hybrid algorithm

▷ **Step** 0 **(Initialization)**
Let $\bar{\epsilon}_1$ and $\bar{\epsilon}_2$ be two positive tolerances for switching from the enumerative method to the semi-smooth. Apply Step 0 of Algorithm 1 and let $\epsilon_1 < \bar{\epsilon}_1$ and $\epsilon_2 < \bar{\epsilon}_2$, where ϵ_1 and ϵ_2 are the tolerances used in Algorithm 1. Let *nmaxit* be the maximum number of iterations allowed to be performed by the semi-smooth method (whenever it is called).

▷ **Step** 1 **(Choice of node)**
Apply Step 1 of Algorithm 1.

▷ **Step** 2 **(Decision step)**
Let $(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ be the stationary point associated with the selected node k and compute θ_1 and θ_2 in (5.1) and (5.2), respectively.

(i) If $\theta_1 \leq \epsilon_1$ and $\theta_2 \leq \epsilon_2$ stop with a solution of SOCQEiCP.

(ii) If $\theta_1 \leq \bar{\epsilon}_1$ and $\theta_2 \leq \bar{\epsilon}_2$ then apply Algorithm 2. If Algorithm 2 terminates with a solution $(x^*, y^*, w^*, t^*, \lambda^*)$ then stop and set $\widetilde{\lambda} = \lambda^*$ and $\widetilde{x} = x^*$. Otherwise, Algorithm 2 terminates without success ($GJ(\widetilde{x}, \widetilde{y}, \widetilde{w}, \widetilde{t}, \widetilde{\lambda})$ is singular or the number of iterations is equal to *nmaxit*); go to Step 2(*iii*).

(iii) Apply Steps 2 and 3 of Algorithm 1 by continuing with the node k and the solution $(\widetilde{x}, \widetilde{y}, \widetilde{v}, \widetilde{w}, \widetilde{\lambda}, \widetilde{z})$ given at the beginning of this step, but skip Step 2(i) and the last instruction of Step 3 of Algorithm 1. Return to Step 1.

8 Computational Experience

In this section, we discuss the numerical performance of the proposed algorithms for computing quadratic complementary eigenvalues. The algorithms have been implemented in MATLAB [24] and the IPOPT (Interior Point OPTimizer) solver [35] has been used to find a (local) solution to the nonlinear problem $\mathbf{NLP}_3(k)$ in (4.19) at each node k.

For the first class of problems, say Test Problems 1, the matrices A and $-C$ were both chosen as the identity matrix, while the matrix B was randomly generated with elements uniformly distributed in the intervals $[0, 1]$, $[0, 5]$, $[0, 10]$, and $[0, 20]$. For these preliminary test problems we have taken $r = 1$. These problems are denoted by RAND$(0, m, n)$, where 0 and m are the end-points of the interval, and n represents the dimension of the problem, i.e., of the matrices A, B, $C \in \mathbb{R}^{n\times n}$. We have considered for generating B, $n = 5$, 10, 20, 30, 40, and 50. Each SOCQEiCP was suitably scaled by using the arguments in Section 2 and with the normalization constraint $\sum_{k=1}^{r} x_0^k = 1$. We solved Test Problems 1 by using the algorithms presented in this paper and the commercial software BARON in order to compare their computational efficiency. The values of tolerances used in the enumerative algorithm and the semi-smooth method were set by considering our experience for the proposed algorithms, while BARON was used with default options.

In Section 8.3, the efficiency of the hybrid algorithm was also investigated for solving more structured problems. First, we analyzed Test problems 1 with $r > 1$, then K is the Cartesian product of Lorentz cones K_i as in (1.10).

We also considered another set of test problems where the matrix A is generated as

$$A = \mu I + G, \tag{8.1}$$

with G being a randomly generated matrix having elements uniformly distributed in the interval $[1, 10]$ and $\mu > \frac{|\min\{0, \theta\}|}{2}$, where θ is the smallest eigenvalue of $G + G^\top$, so that $A \in$ PD. The matrices B and C were chosen as in Test Problem 1 and $r = 1$. Let us call this set of experiments as Test Problems 2.

Finally, the hybrid algorithm was used to solve SOCQEiCP with a bigger dimension for the matrices A, B, C, in particular $n = [100, 250, 500, 1000]$. The matrices were chosen as in both Test Problems 1 and 2 and $r = 1$.

8.1 Performance of the enumerative method

Table 1 reports the computational experience when solving Test Problems 1 with $r = 1$. The enumerative method was run with the tolerances $\epsilon_1 = 10^{-5}$ and $\epsilon_2 = 10^{-5}$. In this table, we report the computed value of the eigenvalue, the value of the function f derived at the solution, the value of the lower and upper bounds for λ computed as in Sections 4.3 and 4.2, respectively, the number of nodes enumerated by the algorithm, and the CPU time in seconds. Furthermore, the column titled "Fe" reports the value of $\|w - \lambda^2 Ax - \lambda Bx - Cx\|_\infty$ derived at the solution, while the last column titled "compl" shows the value of $x^\top w$ at this solution. The value zero in the column titled "Nodes" indicates that a solution to SOCQEiCP was found at the root node itself. The symbol * indicates that the enumerative algorithm was not able to solve the problem, i.e., the algorithm attained the maximum number of iterations, fixed as $n_{\max} = 300$. In this case we include the value of the objective function, the corresponding value of "Fe", and "compl" for the best stationary point available at termination.

As a benchmark for comparison, we solved these same problems using BARON (Branch-And-Reduce Optimization Navigator [34]), which is an optimization solver for the global solution of algebraic nonlinear programs and mixed-integer nonlinear problems. This software package implements a branch-and-reduce algorithm enhanced with a variety of constraint propagation and duality techniques for reducing ranges of variables in the course of the algorithm. The code for solving the nonlinear problem $\mathbf{NLP}_1$ given in (3.3) was implemented in the General Algebraic Modeling Systems (GAMS) language [7] and the solver BARON was used with default options. The numerical results for solving the same set of test problems as above are displayed in Table 2. We use the notation * to indicate that BARON was not able to find a solution to SOCQEiCP.

Comparing Tables 1 and 2, we see that the enumerative method terminated prematurely with just an approximate global optimizer for five test problems, while BARON failed to find a global minimum for nine instances. The values of "Fe" and "compl" obtained with the application of the enumerative algorithm are similar, in general, to those delivered by the global minima given by BARON. Moreover, the computational time for the enumerative method was comparable to that required by BARON.

8.2 Performance of the semi-smooth method

Test Problems 1 with $r = 1$ were solved by using the semi-smooth Newton algorithm presented in Section 6 and the results are shown in Table 3. The starting point was chosen as $\lambda = 1$, $(x^0, \bar{x}, y^0, \bar{y}) = (1/2, 0, 1/2, 0)$, $w = \lambda^2 Ax + \lambda Bx + Cx$, and $t = \lambda x - y$. The algorithm was run with $\widetilde{\epsilon}_1$ and $\widetilde{\epsilon}_2$ both equal to 10^{-4}. In Table 3, we report the value of the computed eigenvalue, the number of iterations taken by the algorithm to converge, and the CPU time in seconds. The notation "*" indicates that the algorithm was not able to converge within the maximum number of iterations, which was set at 100. Note that the

Problem	λ	f	l	u	Nodes	CPU	Fe	compl
RAND(0, 1, 5)	1.082938	4.26029e-09	0.020000	35.272922	0	2.34870e+00	2.02926e-06	3.96564e-05
RAND(0, 1, 10)	1.593798	6.52343e-11	0.627456	124.253405	0	2.08824e+00	8.70040e-07	2.70959e-06
RAND(0, 1, 20)	1.659763	2.81584e-10	0.553049	427.686658	0	1.84855e+00	2.00214e-06	6.42486e-06
RAND(0, 1, 30)	1.946947	4.98848e-08	0.515724	937.744286	5	4.28787e+01	9.32290e-06	5.27508e-05
RAND(0, 1, 40)	1.706686	5.06694e-08	0.376076	1688.709420	7	9.30902e+01	6.55095e-06	6.76991e-05
RAND(0, 1, 50)	2.074764	5.37378e-08	0.660755	2598.493157	11	2.58905e+02	4.25443e-06	4.89964e-05
RAND(0, 5, 5)	3.460789	4.95718e-09	0.396632	77.997883	0	1.71447e+00	2.68093e-06	5.76957e-06
RAND(0, 5, 10)	1.523588	2.24121e-09	0.211826	331.050776	0	2.96092e+00	1.33716e-06	1.75369e-05
RAND(0, 5, 20)	2.812431	2.68931e-07	0.108645	1220.999048	11	6.63263e+01	2.01636e-06	6.22590e-05
RAND(0, 5, 30)	8.890165	2.42596e-07	0.279609	2834.246323	29	1.88836e+02	1.05060e-05	6.48083e-06
RAND(0, 5, 40)	7.126082	1.48623e-05	0.000002	4919.380520	17	2.15128e+02	3.95135e-07	7.52232e-05
RAND(0, 5, 50)	6.778310	2.30355e-08	0.108923	7658.289831	33	6.82388e+02	2.69855e-06	3.66761e-06
RAND(0, 10, 5)	1.721980	4.55823e-10	0.071138	146.082341	0	5.04101e+00	9.90465e-07	6.38694e-06
RAND(0, 10, 10)	*	[2.14363e-04]					1.92697e-04	1.75494e-02
RAND(0, 10, 20)	10.831012	1.28806e-06	0.026954	2253.090185	45	3.24989e+02	6.65534e-06	1.02847e-05
RAND(0, 10, 30)	13.028430	4.62255e-09	0.177992	5015.490181	15	1.21843e+02	6.12185e-06	5.10067e-07
RAND(0, 10, 40)	*	[1.50762e-03]					6.12468e-03	1.18574e-01
RAND(0, 10, 50)	13.738982	3.98646e-04	0.000278	13714.150693	67	1.56999e+03	4.50216e-05	1.05689e-04
RAND(0, 20, 5)	16.255630	1.90235e-09	0.317963	267.804999	9	3.63311e+01	2.43221e-06	1.99696e-07
RAND(0, 20, 10)	*	[2.61659e-06]					8.19066e-05	8.79952e-03
RAND(0, 20, 20)	21.691343	6.55340e-08	0.030432	4217.129671	41	3.16184e+02	8.94613e-06	6.82192e-07
RAND(0, 20, 30)	25.043734	4.32816e-06	0.137434	9410.157670	53	7.09780e+02	3.06579e-06	3.42778e-06
RAND(0, 20, 40)	*	[7.78051e-01]					8.06774e-03	2.59614e-03
RAND(0, 20, 50)	*	[2.71665e-04]					3.34161e-02	4.24448e-03

Table 1: Performance of the enumerative method for solving the scaled SOCQEiCP - Test Problems 1 with $r = 1$.

semi-smooth method is much faster than the enumerative algorithm for obtaining a solution, but on the other hand, it is often not able to converge within the given number of iterations.

$\boxed{8.3}$ Performance of the hybrid method

For all the instances of Test Problems 1 for which the enumerative method required more than one node for finding a solution, we applied the hybrid method proposed in Section 7. The values of the tolerances $\bar{\epsilon}_1$ and $\bar{\epsilon}_2$ used to switch from the enumerative method to the semi-smooth Newton method were set to 10^{-1}. For the semi-smooth Newton algorithm, the values of the tolerances to terminate the algorithm were taken as $\widetilde{\epsilon}_1 = 10^{-4}$ and $\widetilde{\epsilon}_2 = 10^{-4}$. The maximum number of iterations for the semi-smooth method was fixed as 100.

Table 4 displays the value of the computed eigenvalue, the number of nodes enumerated by the algorithm, the number of times that the semi-smooth Newton method was called, which we indicate as "Ntime", the CPU time in seconds, and the values of "Fe" and "compl" defined as above.

We observe that the additional use of the semi-smooth Newton method greatly improved the efficiency and efficacy of the enumerative method. Indeed, the algorithm was able to find a solution by enumerating a fewer number of nodes and successfully solved all the test problems.

In Table 5, we report the results obtained for Test Problems 1, considering $n = [30, 40, 50, 100]$ and $r = 5,\ 10$. These numerical results indicate that the performance of the hybrid algorithm does not seem to be much influenced by the number r of cones K_i.

Table 6 shows that the performance of the hybrid method for solving small dimensional Test Problems 2 is quite good. In fact, the algorithm was always able to terminate without branching. We have only included the case of $r = 1$, as similarly to the Test Problems 1 the performance of the algorithm does not seem to deteriorate when there exists more than one cone K_i ($r > 1$). Tables 7 and 8 present the performance of the hybrid method for solving SOCQEiCPs of larger dimension. In Table 7, we use the notation "*" to indicate

Problem	λ	f	CPU	Fe	Compl
RAND(0, 1, 5)	1.082341	4.31470e-10	2.02500e+00	1.30062e-06	1.10085e-05
RAND(0, 1, 10)	1.593563	4.55799e-10	1.50600e+00	1.58997e-06	6.75148e-06
RAND(0, 1, 20)	1.660184	2.16221e-12	4.34000e+00	1.27705e-08	4.36405e-07
RAND(0, 1, 30)	1.942111	6.28458e-11	4.65900e+00	3.23224e-08	1.81879e-06
RAND(0, 1, 40)	1.704470	1.17660e-16	3.73360e+01	5.45797e-10	3.08607e-09
RAND(0, 1, 50)	*				
RAND(0, 5, 5)	3.459575	1.30447e-11	2.24100e+00	1.07787e-06	2.88253e-07
RAND(0, 5, 10)	1.446998	2.99299e-10	1.12800e+00	4.96887e-06	6.29473e-06
RAND(0, 5, 20)	2.710466	6.54488e-15	6.15600e+00	2.51409e-11	1.02480e-08
RAND(0, 5, 30)	8.877550	9.33578e-14	2.99770e+01	2.08784e-10	3.85227e-09
RAND(0, 5, 40)	*				
RAND(0, 5, 50)	*				
RAND(0, 10, 5)	1.718571	3.15284e-11	2.51100e+00	1.50084e-06	1.54437e-06
RAND(0, 10, 10)	4.330785	4.12716e-10	1.04500e+00	7.50345e-07	1.05383e-06
RAND(0, 10, 20)	*				
RAND(0, 10, 30)	13.019492	8.08185e-12	2.24020e+01	5.76955e-09	1.67219e-08
RAND(0, 10, 40)	*				
RAND(0, 10, 50)	*				
RAND(0, 20, 5)	16.260461	1.01362e-12	4.65100e+00	3.12245e-07	-3.51639e-10
RAND(0, 20, 10)	2.940613	1.02493e-11	1.42800e+00	1.69245e-07	3.48378e-07
RAND(0, 20, 20)	*				
RAND(0, 20, 30)	25.225560	4.50830e-14	1.37040e+02	4.04710e-07	2.52569e-08
RAND(0, 20, 40)	*				
RAND(0, 20, 50)	*				

Table 2: Performance of BARON for solving the scaled SOCQEiCP - Test Problems 1 with $r = 1$.

Problem	λ	It	CPU
RAND(0, 1, 5)	1.081800	5	1.56439e-01
RAND(0, 1, 10)	1.593743	7	6.02723e-03
RAND(0, 1, 20)	1.660417	50	8.24431e-02
RAND(0, 1, 30)	1.942184	11	4.53060e-02
RAND(0, 1, 40)	1.704506	16	9.33797e-02
RAND(0, 1, 50)	2.076201	38	3.15395e-01
RAND(0, 5, 5)	3.459636	22	2.22559e-02
RAND(0, 5, 10)	1.494006	8	1.24778e-02
RAND(0, 5, 20)	2.710538	10	2.41175e-02
RAND(0, 5, 30)	*		
RAND(0, 5, 40)	6.978216	66	3.45564e-01
RAND(0, 5, 50)	*		
RAND(0, 10, 5)	1.759891	5	5.53116e-03
RAND(0, 10, 10)	*		
RAND(0, 10, 20)	*		
RAND(0, 10, 30)	*		
RAND(0, 10, 40)	*		
RAND(0, 10, 50)	*		
RAND(0, 20, 5)	*		
RAND(0, 20, 10)	2.944632	11	1.05081e-02
RAND(0, 20, 20)	*		
RAND(0, 20, 30)	*		
RAND(0, 20, 40)	*		
RAND(0, 20, 50)	*		

Table 3: Performance of the semi-smooth Newton method for solving the scaled SOCQEiCP - Test Problems 1 with $r = 1$.

that the hybrid method was not able to find a solution to SOCQEiCP. This occurred due to the inability of the NLP solver IPOPT to compute an SP of large-scale NLPs at some node of the tree generated by the hybrid algorithm. It is important to add that the hybrid algorithm was able to solve all Test Problems 2 of larger dimension and some of the larger dimensional Test Problems 1. As before, we have only considered the simpler case of $r = 1$, as the performance of the hybrid algorithm does not seem to be much influenced by an

Problem	λ	Nodes	Ntime	CPU	Fe	compl
RAND(0, 1, 30)	1.942184	0	1	2.02698e+00	3.08087e-15	-1.64018e-15
RAND(0, 1, 40)	1.704506	0	1	5.62104e+00	1.41935e-14	-1.48770e-13
RAND(0, 1, 50)	2.076201	0	1	5.97575e+00	3.76871e-12	-1.38359e-12
RAND(0, 5, 20)	2.710538	0	1	4.60102e+00	8.92193e-13	-3.58477e-13
RAND(0, 5, 30)	8.877496	0	1	3.87405e+00	5.73297e-10	-5.70326e-11
RAND(0, 5, 40)	6.978216	0	1	7.41294e+00	6.62892e-12	-4.09478e-13
RAND(0, 5, 50)	6.787334	0	1	9.21684e+00	4.59986e-10	-1.08617e-10
RAND(0, 10, 10)	4.330815	0	1	1.49573e+00	6.37987e-10	-4.92289e-11
RAND(0, 10, 20)	10.831012	2	1	9.18055e+00	7.34059e-11	-1.00148e-11
RAND(0, 10, 30)	13.019383	0	1	5.05396e+00	2.96528e-09	-3.96873e-10
RAND(0, 10, 40)	8.349292	0	1	6.21859e+00	3.64334e-11	-3.56923e-12
RAND(0, 10, 50)	13.185873	0	1	6.59344e+00	1.24879e-11	-3.69148e-12
RAND(0, 20, 5)	16.260338	0	1	5.58621e+00	3.84712e-12	-1.37479e-13
RAND(0, 20, 10)	2.944632	0	1	2.49894e+00	1.12554e-08	-2.40425e-09
RAND(0, 20, 20)	21.671241	0	1	4.60443e+00	3.34048e-12	2.17000e-13
RAND(0, 20, 30)	25.225542	0	1	9.70190e+00	6.67380e-12	-1.19377e-12
RAND(0, 20, 40)	26.071054	1	1	3.33318e+01	2.32863e-13	-6.69950e-15
RAND(0, 20, 50)	26.459219	1	2	3.48536e+01	8.51749e-16	-8.67362e-18

Table 4: Performance of the hybrid method for solving the scaled SOCQEiCP - Test Problems 1 with $r = 1$.

	r=5				
Problem	λ	Nodes	Ntime	Fe	compl
RAND(0, 1, 30)	1.507567	0	1	3.78057e-12	-5.97117e-13
RAND(0, 5, 30)	4.618213	0	1	4.66058e-10	-3.06677e-11
RAND(0, 10, 30)	10.242716	1	1	9.50619e-10	-1.92155e-11
RAND(0, 20, 30)	25.079701	0	1	2.63178e-13	-3.94712e-15
RAND(0, 1, 40)	1.669932	0	1	1.69323e-13	-5.22984e-14
RAND(0, 5, 40)	4.237988	0	1	2.98267e-08	-5.01146e-09
RAND(0, 10, 40)	10.240786	1	1	8.65350e-13	-1.71890e-13
RAND(0, 20, 40)	28.774659	0	1	1.03278e-09	-1.31259e-11
RAND(0, 1, 50)	2.100357	0	1	2.05926e-10	-2.39811e-11
RAND(0, 5, 50)	5.280643	0	1	1.16584e-07	-8.44835e-08
RAND(0, 10, 50)	13.966155	0	1	1.46927e-07	-8.74790e-09
RAND(0, 20, 50)	29.397506	0	1	1.25672e-11	-6.36429e-13
RAND(0, 1, 100)	2.470418	0	1	2.78978e-13	-1.44331e-14
RAND(0, 5, 100)	8.471858	11	5	1.66967e-16	-4.33681e-19
RAND(0, 10, 100)	17.613880	9	4	1.58521e-15	-8.02310e-18
RAND(0, 20, 100)	36.270876	13	1	4.39339e-08	-7.37002e-10
	r=10				
RAND(0, 1, 30)	1.745894	0	1	6.47855e-10	-2.63029e-11
RAND(0, 5, 30)	4.793496	1	1	5.99856e-09	-4.52674e-10
RAND(0, 10, 30)	12.447275	3	3	3.92070e-09	-5.77478e-11
RAND(0, 20, 30)	22.839777	9	2	3.28573e-15	-7.34547e-18
RAND(0, 1, 40)	1.621706	0	1	7.87309e-12	-1.09884e-12
RAND(0, 5, 40)	4.774869	3	4	1.42131e-10	-7.10721e-12
RAND(0, 10, 40)	13.427837	0	1	7.02989e-10	-1.20675e-11
RAND(0, 20, 40)	20.303234	5	2	2.64979e-16	-8.67362e-19
RAND(0, 1, 50)	2.025522	0	1	5.06818e-12	-2.51581e-12
RAND(0, 5, 50)	7.659915	1	2	7.59458e-11	-2.07326e-12
RAND(0, 10, 50)	13.128826	1	1	3.36279e-07	-2.16985e-08
RAND(0, 20, 50)	27.394438	3	1	3.86771e-14	-2.46336e-15
RAND(0, 1, 100)	2.294175	3	1	7.68383e-10	-1.76786e-10
RAND(0, 5, 100)	7.728152	5	1	1.93562e-13	-5.22204e-14
RAND(0, 10, 100)	20.95243	9	2	3.25973e-13	-1.51641e-14
RAND(0, 20, 100)	34.565427	9	1	5.42460e-08	-1.62659e-09

Table 5: Performance of the hybrid method for solving some instances of the scaled SOCQEiCP - Test Problems 1 with $r = 5$ and $r = 10$.

increase of the number r of cones K_i.

In summary, we recommend the proposed hybrid algorithm for solving small and medium scale SOCEiCPs. The algorithm also seems to be able to solve larger problems but its efficiency depends on the efficacy of the NLP solver required for the computation of SPs of NLPs associated with the nodes that are generated during the solution process.

Problem	λ	Nodes	Ntime	CPU	Fe	compl
RAND(0, 1, 5)	0.507923	0	1	3.23268e+00	3.39037e-11	-3.91813e-10
RAND(0, 1, 10)	0.436174	0	1	3.76220e+00	9.42866e-09	-6.49900e-09
RAND(0, 1, 20)	0.354956	0	1	6.32724e+00	4.09118e-09	-1.03500e-08
RAND(0, 1, 30)	0.364482	0	1	6.41729e+00	1.12094e-11	-3.00790e-11
RAND(0, 1, 40)	0.350691	0	1	9.06318e+00	6.55479e-13	-9.55104e-13
RAND(0, 1, 50)	0.282274	0	1	8.66157e+00	4.18338e-12	-9.19511e-13
RAND(0, 5, 5)	0.601275	0	0	7.63706e-01	7.45369e-07	2.15762e-05
RAND(0, 5, 10)	0.874923	0	1	1.27381e+00	4.84486e-12	-1.17259e-12
RAND(0, 5, 20)	0.491872	0	1	1.60495e+00	7.69552e-11	-2.39343e-11
RAND(0, 5, 30)	0.560976	0	1	4.13364e+00	2.06329e-12	-3.80205e-12
RAND(0, 5, 40)	0.450423	0	1	6.69110e+00	6.59862e-10	-2.46968e-10
RAND(0, 5, 50)	0.497433	0	1	9.94207e+00	2.74173e-10	-2.42885e-10
RAND(0, 10, 5)	2.439656	0	1	1.23137e+00	4.09995e-08	2.66742e-09
RAND(0, 10, 10)	0.883004	0	1	4.43661e+00	8.65339e-11	-5.91115e-12
RAND(0, 10, 20)	0.763963	0	1	1.09788e+01	2.60727e-12	-2.56096e-12
RAND(0, 10, 30)	0.643874	0	1	5.76569e+00	1.17444e-11	-8.30925e-12
RAND(0, 10, 40)	0.573783	0	1	9.53772e+00	3.43679e-12	-2.06868e-12
RAND(0, 10, 50)	1.019625	0	1	5.49344e+00	1.54778e-11	-1.34171e-11
RAND(0, 20, 5)	0.319510	0	1	4.42838e+00	5.72875e-14	-4.34999e-14
RAND(0, 20, 10)	1.764161	0	1	1.25310e+00	4.37855e-13	-1.19113e-13
RAND(0, 20, 20)	1.061790	0	1	3.04612e+00	7.96641e-14	-6.04378e-14
RAND(0, 20, 30)	1.567029	0	1	3.12999e+00	6.67869e-16	-1.76942e-16
RAND(0, 20, 40)	1.658715	0	1	8.73667e+00	4.83487e-13	-1.56009e-12
RAND(0, 20, 50)	1.184286	0	1	5.10534e+00	6.66613e-10	-1.25037e-10

Table 6: Performance of the hybrid method for solving the scaled SOCQEiCP - Test Problems 2 with $r = 1$.

Problem	λ	Nodes	Ntime	CPU	Fe	compl
RAND(0, 1, 100)	2.604932	0	1	1.00762e+01	8.32667e-16	-4.44089e-16
RAND(0, 1, 250)	3.848789	0	1	1.14371e+02	2.62984e-15	-1.31622e-16
RAND(0, 1, 500)	4.561722	11	1	1.17903e+04	2.85533e-12	-7.75058e-12
RAND(0, 1, 750)	5.985234	29	11	3.03176e+04	6.43388e-10	-2.54498e-10
RAND(0, 1, 1000)	6.730329	10	3	1.25565e+04	2.41115e-10	-9.41466e-11
RAND(0, 5, 100)	8.464880	0	1	5.05103e+01	4.95485e-10	-2.30437e-10
RAND(0, 5, 250)	16.953133	5	1	6.00345e+03	7.72230e-13	-1.47005e-13
RAND(0, 5, 500)	*					
RAND(0, 5, 750)	*					
RAND(0, 5, 1000)	*					
RAND(0, 10, 100)	22.858033	1	2	2.11507e+02	3.69762e-11	-2.45474e-12
RAND(0, 10, 250)	32.963291	5	1	5.04230e+03	4.34037e-13	-3.02377e-14
RAND(0, 10, 500)	*					
RAND(0, 10, 750)	*					
RAND(0, 10, 1000)	*					
RAND(0, 20, 100)	37.362810	5	1	1.55789e+03	2.64661e-09	-8.67570e-11
RAND(0, 20, 250)	66.240649	24	3	2.18431e+04	5.66524e-10	-7.90562e-11
RAND(0, 20, 500)	*					
RAND(0, 20, 750)	*					
RAND(0, 20, 1000)	*					

Table 7: Performance of the hybrid method for solving the scaled SOCQEiCP - Test Problems 1 with $r = 1$ and $n = [100, 250, 500, 750, 1000]$.

9 Conclusions

In this paper, we have investigated the solution of the Second-Order Cone Quadratic Eigenvalue Complementarity Problem, SOCQEiCP(A, B, C), with $A \in$ PD and $C \in S_0'$. By

Problem	λ	Nodes	Ntime	CPU	Fe	compl
RAND(0, 1, 100)	0.229264	0	1	1.42977e+02	6.04752e-10	-2.46965e-09
RAND(0, 1, 250)	0.197502	0	1	9.14067e+02	5.01144e-14	-1.19606e-13
RAND(0, 1, 500)	0.168158	5	2	6.15534e+03	2.16245e-07	-6.46244e-07
RAND(0, 1, 750)	0.166126	4	1	5.42550e+03	3.39948e-07	-1.02726e-06
RAND(0, 1, 1000)	0.146357	0	1	1.36784e+02	2.42940e-08	-1.01640e-07
RAND(0, 5, 100)	0.444799	0	1	3.45683e+01	7.59938e-10	-1.33166e-09
RAND(0, 5, 250)	0.376290	0	1	1.00536e+03	7.99534e-15	-1.09606e-14
RAND(0, 5, 500)	0.390682	7	1	8.36158e+03	3.84081e-09	-6.25557e-09
RAND(0, 5, 750)	0.359883	8	1	9.75217e+03	3.61709e-10	-2.19914e-10
RAND(0, 5, 1000)	0.345396	0	1	1.24156e+02	2.02728e-14	-1.83707e-14
RAND(0, 10, 100)	0.831757	0	1	1.51574e+02	2.45859e-13	4.42008e-15
RAND(0, 10, 250)	0.587958	3	1	4.00979e+03	5.00239e-11	-5.26204e-10
RAND(0, 10, 500)	0.664101	9	1	1.03858e+04	1.07700e-10	-3.84715e-10
RAND(0, 10, 750)	0.663484	11	1	1.31397e+04	1.26539e-09	-1.52694e-09
RAND(0, 10, 1000)	0.649942	0	1	6.19666e+01	2.22045e-16	-8.32667e-17
RAND(0, 20, 100)	1.342894	0	1	7.41844e+01	2.48597e-15	-9.99201e-16
RAND(0, 20, 250)	1.290551	5	1	6.03008e+03	3.85687e-11	-6.20741e-11
RAND(0, 20, 500)	1.272561	12	2	1.37905e+04	6.57615e-09	-9.66918e-09
RAND(0, 20, 750)	1.342759	10	1	1.14891e+04	6.95174e-13	-6.90838e-13
RAND(0, 20, 1000)	1.268114	0	1	1.56523e+02	1.00533e-12	-1.26593e-12

Table 8: Performance of the hybrid method for solving the scaled SOCQEiCP - Test Problems 2 with $r = 1$ and $n = [100, 250, 500, 750, 1000]$.

exploiting the equivalence between the n-dimensional SOCQEiCP and a suitable $2n$-order SOCEiCP, we introduced an appropriate Nonlinear Programming (NLP) formulation for the latter having a known global optimal value. An enumerative method was developed for solving this NLP formulation and was proven to globally converge to a solution of the SOCQEiCP. However, for some test problems, the enumerative method was able to compute only an approximate solution in practice. Hence, a hybrid method that combines the enumerative algorithm with a semi-smooth method was proposed for implementation, and numerical results were presented to demonstrate that this hybrid method is quite efficient for solving SOCQEiCP.

References

[1] S. Adly and H. Rammal, A new method for solving second-order cone eigenvalue complementarity problem, *J. Optim. Theory Appl.* 165 (2015) 563–585.

[2] S. Adly and S. Seeger, A nonsmooth algorithm for cone constrained eigenvalue problems, *Comp. Optim. Appl.* 49 (2011) 299–318.

[3] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, New York, 2006.

[4] C. Brás, A.N. Iusem and J. Júdice, On the quadratic eigenvalue complementarity problem, *J. Global Optim.* 66 (2016) 153–171.

[5] C. Brás, M. Fukushima, A.N. Iusem and J. Júdice, On the quadratic eigenvalue complementarity problem over a general convex cone, *Applied Math. Comp.* 271 (2015) 594–608.

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

[7] A. Brooke, D. Kendrik, A. Meeraus and R. Raman, *GAMS: A Users' Guide*. GAMS Development Corporation, Washington, 1998.

[8] F.H. Clarke, Optimization and Nonsmooth Analysis *Classics in Applied Mathematics, Society for Industrial and Applied Mathematics*, 1990.

[9] R.W. Cottle, J.S. Pang and R.S. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[10] F. Facchinei and J.S. Pang, *Finite-dimensional Variational Inequalities and Complementarity Problems*, Springer, Berlin, 2003.

[11] U. Faraut, and A. Korányi, *Analysis on Symmetric Cones*, Oxford University Press, New York, 1994.

[12] L.M. Fernandes, M. Fukushima, J. Júdice and H.D. Sherali, The second-order cone eigenvalue complementarity problem. *Optim. Methods Software* 31 (2016) 24–52.

[13] L.M. Fernandes, J. Júdice, M. Fukushima and A. Iusem, A. On the symmetric quadratic eigenvalue complementarity problem, *Optim. Methods Software* 29 (2014) 751–770.

[14] L.M. Fernandes, J. Júdice, H.D. Sherali and M.A. Forjaz, On an enumerative algorithm for solving eigenvalue complementarity problems, *Comp. Optim. Appl.* 59 (2014) 113–134.

[15] L.M. Fernandes, J. Júdice, H.D. Sherali and M. Fukushima, On the computation of all eigenvalues for the eigenvalue complementarity problems. *J. Global Optim.* 59 (2014) 307–326.

[16] M. Fukushima, Z.-Q. Luo and P. Tseng, Smoothing functions for second-order cone complementarity problems, *SIAM J. Optim.* 12 (2001) 436–460.

[17] S. Hayashi, N. Yamashita and M. Fukushima, A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. Optim.* 15 (2005) 593–615.

[18] A.N. Iusem, J. Júdice, V. Sessa and H.D. Sherali, On the numerical solution of the quadratic eigenvalue complementarity problem, *Numerical Algorithms* 72 (2016) 721–747.

[19] J. Júdice, M. Raydan, S. Rosa and S. Santos, On the solution of the symmetric complementarity problem by the spectral projected gradient method, *Numerical Algorithms* 44 (2008) 391–407.

[20] J. Júdice, H.D. Sherali and I. Ribeiro, The eigenvalue complementarity problem. *Comp. Optim. Appl.* 37 (2007) 139–156.

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

[22] H. Le Thi, M. Moeini, T. Pham Dinh and J. Júdice, A DC programming approach for solving the symmetric eigenvalue complementarity problem, *Comp. Optim. Appl.* 5 (2012) 1097–1117.

[23] M.S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, Applications of second-order cone programming. *Linear Algeb. Appl.* 284 (1998) 193–228.

[24] MATLAB, version 8.0.0.783 (R2012b). The MathWorks Inc., Natick, MS (2012).

[25] Y.S. Niu, T. Pham, H.A. Le Thi and J. Júdice, Efficient DC programming approaches for the asymmetric eigenvalue complementarity problem, *Optim. Methods Software* 28 (2013) 812–829.

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

[27] A. Pinto da Costa, J. Martins, I. Figueiredo and J. Júdice, The directional instability problem in systems with frictional contact, *Comp. Meth. Appl. Mec. Eng.* 193 (2004) 357–384.

[28] A. Pinto da Costa and A. Seeger, Cone constrained eigenvalue problems, theory and algorithms, *Comp. Optim. Appl.* 45 (2010) 25–57.

[29] M. Queiroz, J. Júdice and C. Humes, The symmetric eigenvalue complementarity problem, *Math.Comp* 73 (2003) 1849–1863.

[30] H.D. Sherali and W.P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publishers, Dordrecht, 1999.

[31] A. Seeger, Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions. *Linear Algreb. Appl.* 294 (1999) 1–14.

[32] A. Seeger, Quadratic eigenvalue problems under conic constraints, *SIAM J. Matrix Anal. Appl.* 32 (2011) 700–721.

[33] A. Seeger and M. Torky, On eigenvalues induced by a cone constraint, *Linear Algeb. Appl.* 372 (2003) 181–206.

[34] M. Tawarmalani and N.V.A. Sahinidis, A polyhedral branch-and-cut approach to global optimization. *Math. Program.* 103 (2005) 225–249.

[35] A. Wächter and L.T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, *Math. Program.* 106 (2006) 25–57.

Manuscript received 13 October 2014
revised 14 August 2015
accepted for publication 29 January 2016

Alfredo N. Iusem
Instituto de Matemática Pura e Aplicada (IMPA)
Estrada Dona Castorina 110, Rio de Janeiro
RJ, CEP 22460-320, Brazil
E-mail address: iusp@impa.br

Joaquim J. Júdice
Instituto de Telecomunica õ es
Portugal. In the scope of R&D Unit 5000
E-mail address: joaquim.judice@co.it.pt

Valentina Sessa
Janerio (UERJ), Rua São Francisco Xavier, 524, Maracanã
E-mail address: sessa.valentina@gmail.com

Hanif D. Sherali
Grado Department of Industrial, Systems Engineering
Virginia Tech, Blacksburg, VA, USA