



HAL
open science

A two-phase sequential algorithm for global optimization of the standard quadratic programming problem

Joaquim Júdice, Valentina Sessa, Masao Fukushima

► To cite this version:

Joaquim Júdice, Valentina Sessa, Masao Fukushima. A two-phase sequential algorithm for global optimization of the standard quadratic programming problem. 2024. hal-04560413

HAL Id: hal-04560413

<https://hal.science/hal-04560413v1>

Preprint submitted on 26 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A two-phase sequential algorithm for global optimization of the standard quadratic programming problem

Joaquim Júdice · Valentina Sessa · Masao Fukushima

Received: date / Accepted: date

Abstract We introduce a new sequential algorithm for the Standard Quadratic Programming Problem (StQP), which exploits a formulation of StQP as a Linear Program with Linear Complementarity Constraints (LPLCC). The algorithm is finite and guarantees at least in theory a δ -approximate global minimum for an arbitrary small δ , which is a global minimum in practice. The sequential algorithm has two phases. In Phase 1, Stationary Points (SP) with strictly decreasing objective function values are computed. Phase 2 is designed for giving a certificate of global optimality for the last SP computed in Phase 1. Two different Nonlinear Programming Formulations for LPLCC are proposed for each one of these phases, which are solved by efficient enumerative algorithms. New procedures for computing a lower bound for StQP are also proposed, which are easy to implement and give tight bounds in general. Computational experiments with a number of test problems from known sources indicate that the two-phase sequential algorithm is, in general, efficient in practice. Furthermore, the algorithm seems to be an efficient way to study the copositivity of a matrix by exploiting an StQP with this matrix.

Keywords Quadratic Programming · Complementarity Problems · Global Optimization · Maximum Clique Problem

1 Introduction

The Standard Quadratic Programming (StQP) problem consists of finding a (global) minimizer of a quadratic form over the standard simplex. Hence, it can be written in the following form:

$$\text{StQP} : \begin{cases} \min f(y) = y^T Q y \\ \text{s.t. } y \in \Delta, \end{cases} \quad (1)$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric matrix and Δ is the standard simplex in a Euclidean space \mathbb{R}^n , i.e.,

$$\Delta = \{y \in \mathbb{R}^n : y \geq 0, e^T y = 1\}, \quad (2)$$

with e being the n -dimensional vector of ones. StQP (1) has an optimal solution, as its objective function is continuous and its feasible set is nonempty and compact. Note that a general standard quadratic

J. Júdice
Instituto de Telecomunicações, Universidade de Coimbra, Portugal.

V. Sessa*
MINES ParisTech, PSL Research University, CMA Centre de Mathématiques Appliquées, Sophia Antipolis, France.
*corresponding author. E-mail: sessa.valentina@gmail.com

M. Fukushima
The Kyoto College of Graduate Studies for Informatics, Kyoto, Japan.

programming problem with objective function $y^T Ay + 2d^T y$ can be recast into the formulation (1) with $Q = A + de^T + ed^T$.

The name was coined by Bomze in [3], where it is shown that a wide number of applications can be recast as an StQP [5–10, 30], including the so-called Maximum Clique Problem that is quite relevant in computer vision, pattern recognition and robotics [32], portfolio selection problems [1], and testing matrix copositivity [6].

Finding a global optimal solution to StQP (1) when Q is an indefinite matrix, or even giving a certificate of global optimality for a locally computed solution, is known to be NP-hard [24].

Branch-and-Bound (B&B) methods [4] are the algorithms that have been employed for computing a global minimum of an StQP. These algorithms require efficient techniques to find good lower bounds; see [9] and the references therein. In [35], an improved B&B method is proposed. This new version differs from the classical one on the possibility of not expanding some nodes with upper bounds smaller than a dynamically computed target value. This algorithm is applied to the maximum clique problem, and the lower bounds for the algorithm are computed by exploiting the graph structure of the problem. In [12], the B&B method is improved by the use of a combination of reformulation-linearization constraints and also cuts coming from Motzkin-Straus clique theorem [30]. This algorithm is applied to specific classes of graphs with known clique numbers. In [32], the computation of a global minimizer of the StQP is performed by an implicit enumeration of all the cliques of the convexity graph associated to the objective function $f(y)$. Another B&B algorithm is proposed in [27], where an implicit enumeration of Karush-Kuhn-Tucker (KKT) points associated with the StQP is performed based on the property of the convexity graph associated with the StQP. The authors also provide a comparison of two new lower bounds with the best ones computed in [9]. In [20], the KKT conditions associated with StQP (1) are reformulated as a mixed-integer linear programming constraints and StQP is reduced to an equivalent Mixed Integer Linear Program. This program may be solved by well-known solvers, such as CPLEX. Different mixed-integer formulations are compared and an extensive computational experience is reported in that paper.

In this paper, we propose a two-phase sequential algorithm for computing a global minimum of StQP (1). This algorithm exploits the reformulation of StQP as the following Linear Program with Linear Complementarity Constraints (LPLCC):

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & w = Qy - \lambda e \\ & 0 = e^T y - 1 \\ & y \geq 0, w \geq 0 \\ & y^T w = 0. \end{aligned} \tag{3}$$

This reformulation is a direct consequence of the KKT conditions associated to a global minimum of the StQP. The main and new idea of this paper is to separate in two phases the computation of a global minimum of the StQP from the certificate that assures that the possible global minimum computed in the first phase is in fact a global minimum.

In the so-called Phase 1, a number of Stationary Points (SP) of the StQP with strictly decreasing values of the objective function $f(y)$ are sequentially obtained. Each one of these SPs is a solution of the following parametric General Linear Complementarity Problem (GLCP $_u$):

$$\begin{aligned} w &= Qy - \lambda e \\ 0 &= e^T y - 1 \\ y &\geq 0, w \geq 0 \\ y^T w &= 0 \\ \lambda &\leq u. \end{aligned} \tag{4}$$

The parameter u is updated by the formula

$$u = f(\bar{y}) - \alpha \max\{1, |f(\bar{y})|\}, \tag{5}$$

where \bar{y} is the computed SP before the update and α is a small positive number. By using this formula, the computed solutions of the successive GLCP_u lead to SPs with strictly decreasing objective function values. At the end of this sequential procedure, a GLCP_u is found that has no solution. In this case, the last SP \bar{y} associated with the solution of the previous GLCP_u is a δ -approximate global minimum of StQP (1) with

$$\delta = \alpha \max\{1, |f(\bar{y})|\}. \quad (6)$$

In practice, the last computed SP is a global minimum of the StQP provided α is chosen sufficiently small.

An enumerative algorithm is designed to compute a solution of each one of the GLCP_u generated by the sequential procedure. Unfortunately, this algorithm is in general unable to show that the last GLCP_u has no solution. In order to overcome this difficulty, we set a maximum number of iterations for the enumerative algorithm to terminate with a solution. If this number is attained, then it is quite possible that this GLCP_u has no solution and the last computed SP is a δ -approximate global minimum for StQP.

Phase 2 comes to operation to show that the last SP computed in Phase 1 is in fact a δ -approximate global minimum of StQP. A Convex Nonlinear Programming formulation based on the Reformulation Linearisation Technique (RLT) [33] is constructed for the last GLCP_u and either a local nonlinear programming solver in the simplest case or an enumerative algorithm based on this solver is applied to this formulation to show that this GLCP_u has no solution. This gives the required certificate for the last computed SP in Phase 1 to be a δ -approximate global minimum for StQP.

A number of improvements for this two-phase sequential algorithm are proposed. First, two new procedures for computing a lower bound for StQP (1) are introduced. These two techniques are based on diagonal dominance and on Cholesky decomposition and lead to better lower bounds than the popular diagonal lower bound discussed in [9]. It is also shown that StQP (1) can be reduced to an StQP with a strictly copositive matrix by using the computed lower bound. For this new StQP, GLCP_u is shown to be equivalent to a simpler GLCP, which is more appropriate to be used in the two phases of the algorithm. Finally, a Semi-smooth Newton (SN) method can be used in a hybrid method with the enumerative algorithm of Phase 1 in a scheme similar to the one introduced in [17]. This hybrid algorithm is usually more efficient than the simple enumerative algorithm to compute a solution for GLCP_u , when it exists.

Computational experiments with a number of StQP test problems from known sources show that the two-phase sequential algorithm is, in general, efficient for computing a global minimum by finding a sequence of SPs with strictly decreasing objective function values. The new techniques for computing lower bounds usually lead to tight lower bounds that allow the algorithm to give a certificate of global optimality in Phase 1 for many instances. In the remaining cases, Phase 2 is efficient to give such a certificate when a global minimum is in fact computed in Phase 1. For some hard instances, Phase 1 is not able to compute an SP, that is a global minimum, and Phase 2 struggles, as it is unable to compute a better SP. This is not surprising, as Phase 2 is not designed to compute an SP but instead to give a certificate of global optimality for the last SP computed in Phase 1. Actually, this confirms our claim that StQP should be processed in two separate phases. The algorithm has also been shown to be an efficient way to establish whether a symmetric matrix is copositive or not by processing an StQP with this matrix and exploiting some properties of copositive matrices. To have a better idea of the efficiency of the two-phase sequential algorithm in practice, we have solved all the instances with a B&B method [27], which is considered to be the best state-of-art algorithm for StQP. Numerical results reported in this paper indicate that the two-phase sequential algorithm is, in general, competitive with and in some cases superior over the B&B algorithm for computing a global minimum for an StQP and giving a certificate for such global minimum and also for giving a certificate of copositivity for a symmetric matrix.

The structure of the paper is as follows. The sequential algorithm is introduced in Section 2. Section 3 is devoted to the new procedures for computing a lower bound for StQP (1). In Section 4, the reduction of StQP (1) to an StQP with a strictly copositive matrix is discussed. The hybrid enumerative algorithm is presented in Section 5. Phase 2 is discussed in Section 6, namely the RLT formulation and the algorithms for giving a certificate of global optimality based on this formulation. The steps of the two-phase sequential algorithm are presented in Section 7, Computational experience with a number of StQP test problems and matrices to be tested for copositivity is reported in Section 8. Finally, some conclusions and hints for future research are presented in the last section of the paper.

1.1 Notation and properties

We introduce some notation and properties of the StQP that we will use in the sequel.

Let \mathbb{R}^n , \mathbb{R}_+^n and \mathcal{S}^n denote the n -dimensional Euclidean space, the nonnegative orthant and the space of $n \times n$ real symmetric matrices, respectively.

For a vector $x \in \mathbb{R}^n$, we denote its j -th component by x_j , $j \in N$, where N is the index set of x , i.e., $N = \{1, 2, \dots, n\}$. The element (i, j) of a matrix $A \in \mathbb{R}^{n \times n}$ is denoted as a_{ij} .

Let $A \in \mathbb{R}^{n \times n}$. We use $B = \text{diag}(A)$ to indicate a diagonal matrix whose diagonal elements are those of A .

Let $A, B \in \mathbb{R}^{n \times n}$. We use the notation $A \geq B$ whenever $a_{ij} \geq b_{ij}$ for all $i, j \in N$.

The j -th standard basis vector in \mathbb{R}^n is denoted by e^j . We denote by $e \in \mathbb{R}^n$ the n -dimensional vector of ones and by $E = ee^T \in \mathcal{S}^n$ the matrix with all elements being equal to one.

We define the following convex subcones of \mathcal{S}^n :

$$\begin{aligned} \mathcal{N} &= \{A \in \mathcal{S}^n : a_{ij} \geq 0, \forall i, j \in N\} \\ \mathcal{SPSD} &= \{A \in \mathcal{S}^n : x^T A x \geq 0, \forall x \in \mathbb{R}^n\} \\ \mathcal{SPD} &= \{A \in \mathcal{S}^n : x^T A x > 0, \forall x \in \mathbb{R}^n \setminus \{0\}\} \\ \mathcal{COP} &= \{A \in \mathcal{S}^n : x^T A x \geq 0, \forall x \in \mathbb{R}_+^n\} \\ \mathcal{SCOP} &= \{A \in \mathcal{S}^n : x^T A x > 0, \forall x \in \mathbb{R}_+^n \setminus \{0\}\} \end{aligned}$$

that is, the cones of Symmetric Nonnegative, Positive Semi-Definite, Positive Definite, Copositive and Strictly Copositive matrices, respectively.

Testing the copositivity of a given matrix can be formulated as an StQP according to the following property, which is a direct consequence of the definitions of copositive and strictly copositive matrices:

Theorem 1 *Let $Q \in \mathcal{S}^n$ and \bar{y} be a global minimum of StQP (1). Then*

- (i) $Q \in \mathcal{COP}$ if and only if $f(\bar{y}) \geq 0$.
- (ii) $Q \in \mathcal{SCOP}$ if and only if $f(\bar{y}) > 0$.
- (iii) $Q \notin \mathcal{COP}$ if and only if there exists an $y \in \Delta$ such that $f(y) < 0$.
- (iv) $Q \notin \mathcal{SCOP}$ if and only if there exists an $y \in \Delta$ such that $f(y) \leq 0$.

Furthermore, the following properties are used later in this paper:

Theorem 2 [9]

- (i) Let $A \in \mathcal{S}^n$. Let f^* and g^* be the optimal values of StQP (1) with Q replaced by A and $A + \mu E$, respectively. Then,

$$g^* = f^* + \mu$$

for all $\mu \in \mathbb{R}$.

- (ii) Let $Q_1, Q_2 \in \mathcal{S}^n$, and f_1^* and f_2^* be the optimal values of StQP (1) with Q replaced by Q_1 and Q_2 , respectively. If $Q_2 - Q_1 \in \mathcal{N}$, then $f_1^* \leq f_2^*$.

2 A sequential algorithm

We propose a sequential algorithm for finding a global minimum of StQP, which solves a sequence of generalized linear complementarity problems. The steps of the sequential algorithm are presented in Algorithm 1.

Algorithm 1 : Sequential algorithm for StQP

▷ **Step 0: Initialization**

- 1: Let α be a given positive and small number (usually $\alpha = 10^{-3}$).
- 2: Compute a lower bound l for StQP (1).
- 3: Compute an SP \bar{y} for StQP (1).

▷ **Step 1: Update u**

- 4: Compute $f(\bar{y})$ and let u be given by (5).
- 5: If $u - l \leq 0$, then terminate.

▷ **Step 2: Compute a new SP or terminate**

- 6: Find a solution $(\tilde{y}, \tilde{\lambda})$ of GLCP_u (4) or show that GLCP_u has no solution.
 - 7: **if** (GLCP_u has no solution) **then**
 - 8: \tilde{y} is a δ -approximate global minimum of StQP (1), where δ is given by (6); terminate.
 - 9: **else**
 - 10: go to Step 1 with $\bar{y} = \tilde{y}$.
 - 11: **end if**
-

Next, we discuss some important issues about this algorithm.

- (i) A lower bound l for StQP (1) should be computed in the initialization step, as the sequential algorithm stops prematurely at Step 1 with a δ -approximate global minimum \bar{y} if $u \leq l$. A number of techniques have been discussed in [9] for computing a lower bound for StQP. The so-called Diagonal Lower Bound is, in general, the one to be used in practice, as it is simple to implement and gives reasonable good lower bounds for many instances. In Section 3, we introduce new procedures for such a goal. These techniques maintain the simplicity of the diagonal lower bound. Furthermore, they provide a lower bound that is shown to be in theory at least as good as the diagonal lower bound and is in general better for almost all the instances. This issue is discussed in the next section.
- (ii) In the initialization step, a local solver such as **FilterSD** [18] is used to compute an SP of StQP (1). The computed SP depends on the initial point y^0 that is used by the local solver. Two obvious choices are the barycenter of the simplex $y^0 = 1/ne$ and $y^0 = e^r$, where

$$e^r = \operatorname{argmin}\{f(e^i) : i = 1, \dots, n\}, \quad (7)$$

where $e^i, i = 1, \dots, n$, are the standard basis vectors in \mathbb{R}^n . Another good choice is to set y^0 as the vector that is used to compute the lower bound l . If y^t, y^r and y^s are the three SPs computed by the local solver with these three initial points, then the SP \bar{y} computed in the initialization step is given by

$$\bar{y} = \operatorname{argmin}\{f(y^t), f(y^r), f(y^s)\}. \quad (8)$$

- (iii) GLCP_u is an NP-hard problem [24]. Then only an enumerative algorithm is able, in general, to efficiently process it. This approach is fully discussed in Section 5. The algorithm is usually efficient for computing a solution to GLCP_u when it exists, but faces too many difficulties to show that GLCP_u does not have a solution.
- (iv) Showing that the last GLCP_u investigated by the sequential method has no solution gives a certificate for the last computed SP to be a δ -approximate global minimum for StQP (1). Since the enumerative algorithm is unable to terminate in reasonable time with such a certificate, we stop it after a given maximum number of iterations. If this number is attained, the sequential algorithm terminates the so-called Phase 1 with an SP that has a good chance for being a δ -approximate global minimum. Then the algorithm moves to Phase 2, which looks for a certificate of global optimality for this SP. An RLT convex nonlinear programming formulation of GLCP_u is constructed. A nonlinear programming solver in the simplest case or an enumerative algorithm is applied to this optimization problem in order to get the required certificate. This issue is discussed in Section 6.
- (v) We show that StQP (1) can be reduced to an StQP with a \mathcal{SCOP} matrix, by using the lower bound computed in the initialization step. Then, we can apply a new version of Algorithm 1 to this new StQP instead of StQP (1). This improvement is justified by the fact that GLCP_u is shown to be equivalent to a simpler GLCP when the matrix of StQP is \mathcal{SCOP} . The whole issue is fully discussed in Section 4.

- (vi) The sequential nature of Algorithm 1 seems to be interesting for showing that a matrix Q is not \mathcal{COP} (not \mathcal{SCOP}). In fact, it is sufficient to compute an SP \bar{y} of StQP (1) such that $f(\bar{y}) < 0$ ($f(\bar{y}) \leq 0$) to achieve this goal. So, it is not necessary to give a certificate of global optimality in this case and Phase 2 is not required. On the other hand, showing that a matrix Q is \mathcal{COP} (\mathcal{SCOP}) is harder to do, as it is necessary to compute an SP \bar{y} such that $f(\bar{y}) \geq 0$ ($f(\bar{y}) > 0$) and give a certificate of global optimality for this SP. So, the two phases of the algorithm are usually required.

3 Computing a lower bound for StQP

In this section, we present several procedures for computing a lower bound for StQP (1). These are designed by using a shifted nonnegative matrix H and applying Theorem 2(i). Then, an SPD or an SPSD matrix A is derived from this H , such that $A \leq H$. Finally, a lower bound for the StQP (1) is obtained by globally solving a convex StQP (1) with $Q = A$.

3.1 Reduction to an StQP with a nonnegative matrix

Consider StQP (1) and let

$$l_0 = \min\{q_{ij} : i, j = 1, \dots, n\}. \quad (9)$$

If l_0 is attained at a diagonal element q_{rr} , then e^r is a global minimum of StQP (1). So, we assume that l_0 is attained at an off-diagonal element of Q . Hence, $q_{ii} > l_0$ for all i . Consider the matrix

$$H = Q - l_0 E,$$

where E is the matrix of ones. Then, $H \in \mathcal{N}$, $h_{ii} > 0$ for all $i = 1, \dots, n$, and

$$y^T Q y = l_0 + y^T H y. \quad (10)$$

The procedures to be described in this section for computing a lower bound l for StQP (1) start by considering the matrix H as follows:

$$H = \begin{cases} Q, & \text{if } l_0 \geq 0 \\ Q - l_0 E, & \text{otherwise.} \end{cases} \quad (11)$$

Then, a lower bound \bar{l} is computed for the following StQP:

$$\min_{y \in \Delta} y^T H y. \quad (12)$$

Finally, a lower bound l for StQP (1) can be derived as follows

$$l = \begin{cases} \bar{l}, & \text{if } l_0 \geq 0 \\ l_0 + \bar{l}, & \text{otherwise.} \end{cases} \quad (13)$$

In order to compute \bar{l} , we solve the convex StQP

$$\min_{y \in \Delta} y^T A y, \quad (14)$$

where $A \in \mathcal{SPSD}$ or $A \in \mathcal{SPD}$ such that

$$0 \leq A \leq H. \quad (15)$$

Hence, by Theorem 2(ii), the global optimal value of StQP (14) gives a lower bound \bar{l} for StQP (12). Since $A \in \mathcal{SPSD}$, then StQP (14) is a convex program, that can be easily solved by a local solver.

The main issue is the matrix A that should be simple to construct and should imply that StQP (14) can be solved with a relatively small effort. The simplest choice is to set $A = \text{diag}(H)$. This leads to the very popular Diagonal Lower Bound [9]. Next, we introduce some procedures for constructing the matrix A that provide lower bounds for StQP (12) where A strictly contains the diagonal of H . Hence, by Theorem 2(ii), these techniques provide, in general, better lower bounds than the Diagonal Lower Bound.

3.2 Lower bounds based on diagonal dominance

The following property holds [14]

Theorem 3 *If $A \in \mathcal{N}$ and $a_{ii} \geq \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}$ for all $i = 1, \dots, n$, then $A \in \mathcal{SPSD}$.*

We construct a symmetric nonnegative matrix A satisfying the hypothesis of Theorem 3 and the condition (15). First, we set $a_{ii} = h_{ii}$, for all $i = 1, \dots, n$. Then,

(i) the elements a_{ij} , $j > i$ of the strictly upper triangular part of A are the values of the variables $a_{(ij)}$ of an optimal solution of one of the two following optimization problems:

(a) Linear Program (LP):

$$\begin{aligned} \text{LP : } \quad & \min \sum_{i=1}^n \sum_{j=i+1}^n (h_{ij} - a_{(ij)}) \\ & \text{s.t. } \quad h_{ii} \geq \sum_{\substack{j=1 \\ j \neq i}}^n a_{(ij)}, \quad i = 1, \dots, n \\ & \quad \quad 0 \leq a_{(ij)} \leq h_{ij}, \quad i = 1, \dots, n, \quad j = i + 1, \dots, n. \end{aligned} \tag{16}$$

(b) Convex Quadratic Program (CQP):

$$\begin{aligned} \text{CQP : } \quad & \min \sum_{i=1}^n \sum_{j=i+1}^n (h_{ij} - a_{(ij)})^2 \\ & \text{s.t. } \quad h_{ii} \geq \sum_{\substack{j=1 \\ j \neq i}}^n a_{(ij)}, \quad i = 1, \dots, n, \\ & \quad \quad 0 \leq a_{(ij)} \leq h_{ij}, \quad i = 1, \dots, n, \quad j = i + 1, \dots, n. \end{aligned} \tag{17}$$

(ii) The elements of the strictly lower triangular part of A are given by:

$$a_{ji} = a_{(ij)}, \quad \text{for } j > i.$$

The last constraints in the optimization problems above imply that $A \in \mathcal{SPSD}$ and $0 \leq A \leq H$. These two optimization problems have an optimal solution since the constraint set is convex, compact, and nonempty. They can be efficiently solved by a linear programming or a local nonlinear programming solver, such as **GUROBI** [21]. Furthermore, both LP and CQP have nonzero optimal solutions, which means that A strictly contains the diagonal of H .

3.3 Lower bound based on Cholesky factorization

Next, we introduce another procedure for computing the matrix A , that uses a modification of the so-called Bordering Method for computing the LDL^T decomposition (or Cholesky factorization) of an SPD matrix [29]. It is designed to build a matrix $A \in \mathcal{SPD}$ such that $A_{FF} = H_{FF}$ where F is a set with cardinality $p \leq n$ and all rows $A_{r\bullet}$ (resp., columns $A_{\bullet r}$) for $r \notin F$ are given by $A_{r\bullet} = h_{rr}(e^r)^T$ (resp., $A_{\bullet r} = h_{rr}e^r$), where e^r is the r -th standard basis vector in \mathbb{R}^n . Note that $p = n$ if and only if $H \in \mathcal{SPD}$. The steps of this procedure are presented in Algorithm 2.

Algorithm 2 : Modified Bordering Algorithm

▷ **Step 0: Initialization**

1:

$$\begin{aligned} A &:= H \\ F &:= \{1\} \\ G &:= \{2, \dots, n\} \\ \bar{a}_{11} &= a_{11} \end{aligned}$$

▷ **Step 1: Iterative step**

2: **while** $|G| > 0$ **do**:

3: Let $r = \min\{i \in G\}$ and set $G := G \setminus \{r\}$.

4: Solve for \bar{A}_{Fr}

$$A_{FF} \bar{A}_{Fr} = A_{Fr}$$

5: Compute

$$\bar{a}_{rr} = a_{rr} - A_{Fr}^T \bar{A}_{Fr}$$

If $\bar{a}_{rr} > 0$, **then**

$$F = F \cup \{r\}$$

else

$$a_{rj} = a_{jr} = 0, \quad j \neq r, \quad j = 1, \dots, n$$

$$\bar{a}_{rr} = a_{rr}$$

end if

6: **end while**

The Modified Bordering Method computes the diagonal elements \bar{a}_{ii} , $i = 1, \dots, n$ of the LDL^T decomposition of a matrix A . Since all these elements are positive, then $A \in \mathcal{SPD}$.

At the end of the procedure, we obtain a matrix A that is given by

$$P^T A P = \begin{bmatrix} H_{FF} & 0 \\ 0 & \text{diag}(H_{\bar{F}\bar{F}}) \end{bmatrix}, \quad (18)$$

where P is a permutation matrix and $\bar{F} = \{1, \dots, n\} \setminus F$. Note that $F \neq \emptyset$, which implies that A strictly contains the diagonal of H .

In many cases, the set F may have a small number of elements. Next, we introduce a modification that looks for a matrix A with a set F of bigger cardinality. Let

$$K = \{(i, j) \in N \times N : h_{ij}^2 \geq h_{ii}h_{jj} \text{ and } i \neq j\}, \quad (19)$$

and consider the matrix $\bar{H} = [\bar{h}_{ij}]$, where

$$\bar{h}_{ij} = \begin{cases} h_{ij} & \text{if } (i, j) \notin K \\ h_{ij} - \alpha_{ij} & \text{if } (i, j) \in K \end{cases}, \quad (20)$$

where α_{ij} satisfies

$$0 < \alpha_{ij} < h_{ij}. \quad (21)$$

Then

$$0 \leq \bar{H} \leq H,$$

and

$$\min_{y \in \Delta} y^T \bar{H} y \leq \min_{y \in \Delta} y^T H y.$$

Then, the optimal value of the following StQP

$$\min_{y \in \Delta} y^T \bar{H} y \tag{22}$$

gives a lower bound for StQP (12). Note that, the closer α_{ij} are to zero the closer \bar{H} is to H and the better the lower bound is. In practice, α_{ij} for each $(i, j) \in K$ are computed by

$$\alpha_{ij} = \gamma + h_{ij} - \sqrt{h_{ii}h_{jj}} \tag{23}$$

where $\gamma > 0$.

As before, the matrix A is obtained by applying Algorithm 2 to the matrix \bar{H} .

Numerical experiments showed that the choice of γ is an important issue for the Modified Bordering method to compute a matrix A that is close to \bar{H} . We suggest to use $\gamma = \min\{\sqrt{h_{ii}h_{jj}} : (i, j) \in K\} - 10^{-p} > 0$, with $p \in \{0, 1, 2, 3, 4\}$. An alternative could be to use $\gamma = \gamma_{ij}$, where $\gamma_{ij} = \sqrt{h_{ii}h_{jj}} - 10^{-p} > 0$, with $p \in \{0, 1, 2, 3, 4\}$ for each $(i, j) \in K$.

Computational experiments on a number of test problems indicate that there is not a clear winner when the matrix A is chosen by the two diagonal dominance LP and CQP procedures or by the modified Cholesky method applied to the matrix \bar{H} given by (20). For this reason, we compute the lower bounds by these three procedures and then we chose the largest of the three values to be the lower bound for StQP (1). Furthermore, these experiments also show that the lower bound computed by this procedure is better than the diagonal lower bound for almost all the test problems.

4 An equivalent StQP with a strictly copositive matrix

Consider StQP (1) and let l be a lower bound for this problem. If $l > 0$, then $Q \in SCOP$ and we set $M = Q$. If $l \leq 0$, then we set

$$M = Q - (l - \eta)E, \tag{24}$$

where $\eta > 0$. Then

$$y^T M y \geq \eta > 0, \quad \forall y \in \Delta$$

and $M \in SCOP$. Furthermore,

$$\min_{y \in \Delta} y^T Q y = \min_{y \in \Delta} y^T M y + (l - \eta) \tag{25}$$

So, after the initialization step, the sequential algorithm can be applied to the following StQP with a *SCOP* matrix:

$$\min_{y \in \Delta} f_M(y) = y^T M y, \tag{26}$$

instead of StQP (1). This simply amounts to replace the matrix Q by the matrix M , and the lower bound l by

$$l := \begin{cases} l & \text{if } l > 0 \\ \eta & \text{otherwise.} \end{cases} \tag{27}$$

An optimal solution of StQP (26) satisfies the KKT conditions, which are given by the following Mixed Linear Complementarity Problem (LCP):

$$w = My - \lambda e \quad (28a)$$

$$y \geq 0, w \geq 0 \quad (28b)$$

$$y^T w = 0 \quad (28c)$$

$$e^T y = 1, \quad (28d)$$

where $w \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ are the dual variables associated with the constraints $y \geq 0$ and $e^T y = 1$, respectively, scaled by a factor 1/2. Note that (28a), (28c), and (28d) imply $f_M(y) = \lambda$ in each solution of (28).

Next, we show that a solution of the Mixed LCP (28) can be retrieved from the solution of a Linear Complementarity Problem (LCP), which has one less variable and one less constraint. Consider the LCP

$$v = Mx - e \quad (29a)$$

$$x \geq 0, v \geq 0 \quad (29b)$$

$$x^T v = 0. \quad (29c)$$

Then the following property holds.

Theorem 4 \bar{x} is a solution of LCP (29) if and only if $\left(\bar{y} = \frac{\bar{x}}{e^T \bar{x}}, \bar{\lambda} = \frac{1}{e^T \bar{x}}\right)$ is a solution of Mixed LCP (28).

Proof (i) Let \bar{x} be a solution of LCP (29) and $\bar{v} = -e + M\bar{x}$. Then $0 \neq \bar{x}$, $e^T \bar{x} > 0$ and

$$\begin{aligned} \frac{\bar{v}}{e^T \bar{x}} &= M \frac{\bar{x}}{e^T \bar{x}} - \frac{1}{e^T \bar{x}} e \\ \frac{\bar{x}}{e^T \bar{x}} &\geq 0, \frac{\bar{v}}{e^T \bar{x}} \geq 0 \\ \left(\frac{\bar{x}}{e^T \bar{x}}\right)^T \left(\frac{\bar{v}}{e^T \bar{x}}\right) &= 0 \end{aligned}$$

So $\left(\bar{y} = \frac{\bar{x}}{e^T \bar{x}}, \bar{\lambda} = \frac{1}{e^T \bar{x}}\right)$ is a solution of the Mixed LCP (28).

(ii) Let $(\bar{y}, \bar{\lambda})$ be a solution of the Mixed LCP (28) and $\bar{w} = M\bar{y} - \bar{\lambda}e$. Then

$$\bar{y}^T \bar{w} = 0 = \bar{y}^T M\bar{y} - \bar{\lambda} e^T \bar{y}.$$

So

$$\bar{\lambda} = \bar{y}^T M\bar{y} > 0 \quad (30)$$

as $M \in SCOP$. Moreover, we have

$$\begin{aligned} \frac{\bar{w}}{\bar{\lambda}} &= M \frac{\bar{y}}{\bar{\lambda}} - e \\ \frac{\bar{w}}{\bar{\lambda}} &\geq 0, \frac{\bar{y}}{\bar{\lambda}} \geq 0 \\ \left(\frac{\bar{w}}{\bar{\lambda}}\right)^T \frac{\bar{y}}{\bar{\lambda}} &= 0. \end{aligned}$$

So $\bar{x} = \frac{\bar{y}}{\bar{\lambda}}$ is a solution of LCP (29). □

It follows from the proof of Theorem 4 that for a solution \bar{x} of LCP (29) and a solution $(\bar{y}, \bar{\lambda})$ of Mixed LCP (28), we have

$$\bar{\lambda} = \frac{1}{e^T \bar{x}} > 0. \quad (31)$$

Then, StQP (26) can be reformulated as the following Linear Program with Linear Complementarity Constraints (LPLCC):

$$\begin{aligned} \max \quad & e^T x \\ \text{s.t.} \quad & v = Mx - e \\ & x \geq 0, v \geq 0 \\ & x^T v = 0. \end{aligned} \quad (32)$$

Now, suppose that we apply Algorithm 1 to StQP (26). Then, instead of solving GLCP_u , we solve GLCP_r

$$v = Mx - e \quad (33a)$$

$$x \geq 0, v \geq 0 \quad (33b)$$

$$x^T v = 0 \quad (33c)$$

$$e^T x \geq r, \quad (33d)$$

where

$$r = \frac{1}{u}. \quad (34)$$

The reformulated Algorithm 1 to StQP (26) instead of the original StQP is shown below.

Algorithm 1' : Sequential algorithm for StQP (26)

▷ **Step 0: Initialization**

- 1: Let α be a given positive and small number (usually $\alpha = 10^{-3}$).
- 2: Compute a lower bound l_Q to StQP (1) by the procedure described in Section 3.
- 3: **if** $l_Q \leq 0$ **then**
- 4: set $M = Q - (l_Q - \eta)E$ and $l = \eta$.
- 5: **else**
- 6: set $M = Q$ and $l = l_Q$.
- 7: **end if**
- 8: Compute an SP \bar{y} of StQP (26).

▷ **Step 1: Update u**

- 9: Compute $f_M(\bar{y})$ and let $r = 1/u$, where u is given by

$$u = f_M(\bar{y}) - \alpha \max\{1, |f_M(\bar{y})|\}. \quad (35)$$

- 10: If $u - l \leq 0$, then terminate.

▷ **Step 2: Compute a new SP or terminate**

- 11: Find a solution \bar{x} of GLCP_r (33) or show that GLCP_r has no solution.
 - 12: **if** (GLCP_r has no solution) **then**
 - 13: \bar{y} is a δ -approximate global minimum of StQP (26), where δ is given by (6); terminate.
 - 14: **else**
 - 15: go to Step 1 with $\bar{y} = \bar{x}/e^T \bar{x}$.
 - 16: **end if**
-

In the next section, we propose an enumerative algorithm for solving $GLCP_r$. This algorithm is applied to the following Quadratic Program, QP_r , associated to $GLCP_r$:

$$\min \quad c(x, v) := x^T v \quad (36a)$$

$$\text{s.t.} \quad v = Mx - e \quad (36b)$$

$$x \geq 0, v \geq 0 \quad (36c)$$

$$e^T x \geq r. \quad (36d)$$

Note that QP_r is bounded from below by zero on its feasible set. Therefore, it has a nonnegative global optimal value provided it is feasible. The following properties hold:

Theorem 5 (\bar{x}, \bar{v}) is an SP of QP_r with $c(\bar{x}, \bar{v}) = 0$ if and only if (\bar{x}, \bar{v}) is a solution of $GLCP_r$.

Proof (i) If (\bar{x}, \bar{v}) is an SP of QP_r with $c(\bar{x}, \bar{v}) = 0$, then (\bar{x}, \bar{v}) satisfies all the linear constraints of $GLCP_r$ and the complementarity condition. Hence, (\bar{x}, \bar{v}) is a solution of $GLCP_r$.

(ii) Let (\bar{x}, \bar{v}) a solution of $GLCP_r$. Since the linear constraints of $GLCP_r$ are the constraints of QP_r , then this program is feasible. Since QP_r is bounded from below by zero, then (\bar{x}, \bar{v}) is a global optimal solution of QP_r . Hence, (\bar{x}, \bar{v}) is an SP of QP_r with $c(\bar{x}, \bar{v}) = 0$. □

Theorem 6 (\bar{x}, \bar{v}) is an SP of QP_r with $c(\bar{x}, \bar{v}) = 0$ if and only if $\bar{y} = \bar{x}/e^T \bar{x}$ is an SP of $StQP$ (26) with $f_M(\bar{y}) = \frac{1}{e^T \bar{x}} \leq \frac{1}{r}$.

Proof (i) Let (\bar{x}, \bar{v}) be an SP of QP_r and $c(\bar{x}, \bar{v}) = 0$. Hence, (\bar{x}, \bar{v}) is a solution of LCP (29). By Theorem 4 $(\bar{y} = \frac{\bar{x}}{e^T \bar{x}}, \bar{\lambda} = \frac{1}{e^T \bar{x}})$ is a solution of Mixed LCP (28). Since any solution of this Mixed LCP is an SP of $StQP$ (26), then \bar{y} is an SP of $StQP$ (26) with $f_M(\bar{y}) = \bar{\lambda}$. As $\bar{\lambda} = \frac{1}{e^T \bar{x}}$ and \bar{x} satisfies (36d), then

$$f_M(\bar{y}) \leq \frac{1}{r}.$$

(ii) Let \bar{y} be an SP of $StQP$ (26) such that $f_M(\bar{y}) \leq \frac{1}{r}$. Then, $(\bar{y}, \bar{\lambda} = \bar{y}^T M \bar{y})$ is a solution of Mixed LCP (28) and $\bar{\lambda} \leq \frac{1}{r}$. By Theorem 4, $\bar{x} = \frac{\bar{y}}{\bar{\lambda}}$ is a solution of LCP (29) and $\bar{\lambda} = \frac{1}{e^T \bar{x}}$, which gives $e^T \bar{x} \geq r$. Then, $(\bar{x}, \bar{v} = -e + M\bar{x})$ is a solution of $GLCP_r$ and by Theorem 5 (\bar{x}, \bar{v}) is an SP of QP_r with $c(\bar{x}, \bar{v}) = 0$. □

Theorem 7 $GLCP_r$ has no solution if and only if QP_r is infeasible or has a positive global optimal value.

Proof (i) If $GLCP_r$ has no solution, then either its linear constraints are inconsistent or any feasible solution (\bar{x}, \bar{v}) of $GLCP_r$ satisfies $c(\bar{x}, \bar{v}) > 0$. Hence, either QP_r is infeasible or it has a positive global optimal value.

(ii) If QP_r is infeasible, then $GLCP_r$ is infeasible and has no solution. If QP_r has a positive global optimal value, then there exists no feasible solution (\bar{x}, \bar{v}) such that $c(\bar{x}, \bar{v}) = 0$. Then $GLCP_r$ has no solution. □

5 An enumerative algorithm for Phase 1

The enumerative algorithm to be described in this section explores a binary tree, which is constructed by using a branching strategy based on the complementarity conditions between the variables v_i and x_i , i.e., either $v_i = 0$ or $x_i = 0$ for each $i = 1, \dots, n$ in $GLCP_r$. Each node k of the tree is associated with a QP_r^k defined by QP_r and some variables $x_i = 0$ or $v_i = 0$ that have been fixed to generate this node k . Note that at the root node $k := 1$ and $QP_r^1 = QP_r$. The algorithm computes an SP (\bar{x}, \bar{v}) for this QP_r^k and there are two cases:

- (i) $c(\bar{x}, \bar{v}) = 0$ (i.e., smaller than a tolerance ϵ). Then by Theorem 5, (\bar{x}, \bar{v}) is a solution of GLCP_r and the algorithm stops.
- (ii) $c(\bar{x}, \bar{v}) > 0$ and two nodes are generated from node k by fixing $x_i = 0$ or $v_i = 0$, where the index i is chosen according to some heuristic rule.

The algorithm constructs such a binary tree and either finds a solution of GLCP_r at a certain node or terminates with no more nodes to visit and a message that GLCP_r has no solution. The steps of Algorithm 3 are presented below.

Algorithm 3 : Enumerative algorithm for GLCP_r in Phase 1

▷ Step 0: Initialization

- 1: Set $\epsilon_1, \epsilon_2 > 0$ ($\epsilon_1 = 10^{-4}$, $\epsilon_2 = 10^{-6}$ by default).
- 2: If QP_r is infeasible, then GLCP_r has no solution; terminate.
- 3: Compute an SP (\bar{x}, \bar{v}) of QP_r and let

$$\theta = \max\{\min\{\bar{x}_j, \bar{v}_j\} : j = 1, \dots, n\}. \quad (37)$$

- 4: If $c(\bar{x}, \bar{v}) \leq \epsilon_1$ OR $\theta \leq \epsilon_2$, then terminate: (\bar{x}, \bar{v}) is a solution of GLCP_r and $\bar{y} = \bar{x}/e^T \bar{x}$ is an SP of StQP (26) (and of StQP (1)).
- 5: Otherwise, let $P = \{1\}$ be the set of open nodes and $N = 1$ be the number of nodes generated by the algorithm.
- 6: Set $k = 1$, $\text{UB}(1) = c(\bar{x}, \bar{v})$ and $i(1) = j^*$ with j^* being the index where θ is achieved in (37).

▷ Step 1: Choice of node

- 7: If $P = \emptyset$, terminate: QP_r (36) has a positive global optimal value and GLCP_r has no solution.
- 8: Otherwise, select $k \in P$ such that $\text{UB}(k) = \min\{\text{UB}(i) : i \in P\}$, set $j^* = i(k)$ and $P = P \setminus \{k\}$.
- 9: Branch on the complementary variables \bar{x}_{j^*} and \bar{v}_{j^*} (i.e., fix each one of these variables to zero) and generate two new nodes $N + 1$ and $N + 2$.

▷ Step 2: Solve, Update and Queue

- 10: For $p = N + 1$ and $p = N + 2$, find a stationary point (\bar{x}, \bar{v}) of QP_r^p .
 - 11: If QP_r^p is feasible, then
 - (i) If $c(\bar{x}, \bar{v}) \leq \epsilon_1$ OR $\theta \leq \epsilon_2$, then terminate: (\bar{x}, \bar{v}) is a solution of GLCP_r and $\bar{y} = \bar{x}/e^T \bar{x}$ is an SP of StQP (26) (and of StQP (1)).
 - (ii) Else, set $P = P \cup \{p\}$, $\text{UB}(p) = c(\bar{x}, \bar{v})$ and $i(p) = j^*$, with j^* being the index where θ is achieved in (37).
 - 12: Return to Step 1.
-

Note that if Algorithm 3 terminates in Step 0, then either the current SP of QP_r is a solution of GLCP_r , or QP_r is infeasible and GLCP_r has no solution. In these two simple cases, the algorithm stops without branching, i.e., with a number of nodes equal to zero.

Termination in Step 1 with an empty list of open nodes usually requires too many iterations. So, in practice, a maximum number of iterations niter_{max} is considered. This number should be chosen with care, as a large value implies too much computational work. On the other hand, the computation of a new SP of StQP (26) may be lost if this value is too small. The value $\text{niter}_{max} = 100$ seems to work well in practice and is used by default.

As shown by our numerical experiments, Algorithm 3 is usually efficient to find a new SP of StQP (26), since it has only to compute an SP of a QP_r^k with zero objective function value. On the contrary, the algorithm is usually unable to give a certificate of global optimality for StQP (26) by showing that GLCP_r has no solution.

A semi-smooth Newton (SN) method can also be used to help the computation of a solution of a GLCP_r in a hybrid scheme with the enumerative algorithm that is explained later in this section. In order to use

the SN method, we consider the following LCP that is equivalent to GLCP_r :

$$v = Mx - e \quad (38a)$$

$$v_{n+1} = -r + e^T x - x_{n+1} \quad (38b)$$

$$x \geq 0, v \geq 0, x^T v = 0, \quad (38c)$$

$$x_{n+1} \geq 0, v_{n+1} \geq 0, x_{n+1}^T v_{n+1} = 0 \quad (38d)$$

Consider the Fischer-Burmeister (FB) function $\varphi_{FB}(a, b) = a + b - \sqrt{a^2 + b^2}$ with $a, b \in \mathbb{R}$ to replace each complementarity constraint ($v_i \geq 0, x_i \geq 0, v_i x_i = 0$) by an equation. This means that we write the $n + 1$ complementarity constraints as

$$\begin{bmatrix} \Phi_{FB}(x, v) \\ \Phi_{FB}(x_{n+1}, v_{n+1}) \end{bmatrix} = \begin{bmatrix} \varphi_{FB}(x_1, v_1) \\ \vdots \\ \varphi_{FB}(x_n, v_n) \\ \varphi_{FB}(x_{n+1}, v_{n+1}) \end{bmatrix} = 0. \quad (39)$$

If we use the Fischer-Burmeister function to represent the complementarity constraints, then system (38) can be written as:

$$\Psi(x, x_{n+1}, v, v_{n+1}) = \begin{bmatrix} v - Mx + e \\ x_{n+1} + v_{n+1} + r - e^T x \\ \Phi_{FB}(x, v) \\ \Phi_{FB}(x_{n+1}, v_{n+1}) \end{bmatrix} = 0. \quad (40)$$

In each iteration of the SN method, a direction is computed by considering the following system:

$$Jd = q, \quad (41)$$

with

$$d = \begin{bmatrix} d_x \\ d_{x_{n+1}} \\ d_v \\ d_{v_{n+1}} \end{bmatrix} \text{ and } q = \begin{bmatrix} -\bar{v} + M\bar{x} - e \\ -\bar{x}_{n+1} - \bar{v}_{n+1} - r + e^T \bar{x} \\ -\Phi_{FB}(x, v) \\ -\Phi_{FB}(x_{n+1}, v_{n+1}) \end{bmatrix},$$

where $(\bar{x}, \bar{x}_{n+1}, \bar{v}, \bar{v}_{n+1})$ is the current Newton iterate that satisfies $e^T \bar{x} \geq r$. Moreover, J is the Clarke Generalized Jacobian $J(\bar{x}, \bar{x}_{n+1}, \bar{v}, \bar{v}_{n+1})$ given by

$$J(\bar{x}, \bar{x}_{n+1}, \bar{v}, \bar{v}_{n+1}) = \begin{bmatrix} -M & 0 & I_n & 0 \\ -e^T & 1 & 0 & 1 \\ V & 0 & W & 0 \\ 0 & V_{n+1} & 0 & W_{n+1} \end{bmatrix} \in R^{(2n+2) \times (2n+2)}, \quad (42)$$

where the diagonal elements of the diagonal matrices V and W and the elements V_{n+1} and W_{n+1} are computed as follows:

$$(V_i, W_i) = \begin{cases} \left(1 - \frac{\bar{x}_i}{\sqrt{\bar{x}_i^2 + \bar{v}_i^2}}, 1 - \frac{\bar{v}_i}{\sqrt{\bar{x}_i^2 + \bar{v}_i^2}} \right) & \text{if } (\bar{x}_i, \bar{v}_i) \neq 0 \\ (1 - \xi_i, 1 - \eta_i) & \text{if } (\bar{x}_i, \bar{v}_i) = 0 \end{cases} \quad \forall i = 1, \dots, n + 1, \quad (43)$$

with $\xi_i^2 + \eta_i^2 = 1$. In practice, we use $(\xi_i, \eta_i) = (0, 1)$ for all $i = 1, \dots, n + 1$.

Now there are two cases:

- (i) J is nonsingular and d is the unique solution of system (41).

(ii) J is singular and d is the minimum norm solution of the Linear Least Squares (LSQ) problem

$$\min \|Jd - q\|^2 \quad (44)$$

associated with the system (41). Alternatively, we can compute d by using the Levenberg-Marquardt (LM) formula

$$\min \left\| \begin{bmatrix} J \\ \mu I \end{bmatrix} d - \begin{bmatrix} q \\ 0 \end{bmatrix} \right\|^2, \quad (45)$$

where μ is a real number and I is the identity matrix. Note that the LM formula (45) reduces to the formula (44) when $\mu = 0$.

Then, a new iterate is computed by:

$$\tilde{x} = \bar{x} + d_x, \quad \tilde{x}_{n+1} = \bar{x}_{n+1} + d_{x_{n+1}}, \quad \tilde{v} = \bar{v} + d_v, \quad \tilde{v}_{n+1} = \bar{v}_{n+1} + d_{v_{n+1}}, \quad (46)$$

where the Newton direction d is uniquely determined by solving (41) or (44) (or (45)).

Stopping Criterion:

An iterate (x, v) is a solution of GLCP_r if and only if it satisfies (38a) and (38c), and $e^T x \geq r$. Hence, we suggest the following Stopping Criterion for the SN method:

$$\|v - Mx + e\| \leq \epsilon \quad (47)$$

$$\max\{|\min\{x_i, v_i\}|, i = 1, \dots, n+1\} \leq \epsilon \quad (48)$$

$$e^T x \geq r + \epsilon, \quad (49)$$

where ϵ is a tolerance for zero ($\epsilon = 10^{-6}$ by default).

Note that it is also possible to develop an SN algorithm with line search similar to the one described in [25] in order to overcome the case of the singularity of the Clarke Generalized Jacobian. However, numerical results reported in [25] indicate that this new version of the SN method does not seem to perform better in practice than the simple version described in this section. So, we do not use this version of the SN method with line search in practice.

Initial point:

An initial point (x, x_{n+1}, v, v_{n+1}) for the SN method can be chosen as

$$x = \tilde{x}, \quad v = \tilde{v}, \quad x_{n+1} = 0, \quad \text{and} \quad v_{n+1} = -r + e^T \tilde{x},$$

where (\tilde{x}, \tilde{v}) is the current point used at node k .

Improved Enumerative Algorithm:

Next, we explain how the hybrid scheme works. Suppose that Algorithm 3 is used in Step 2 of Algorithm 1' with a node k and a point (\tilde{x}, \tilde{v}) . If θ given by (37) satisfies $\theta \leq \bar{\epsilon}$ (where $\bar{\epsilon} = \sqrt{\epsilon_2}$), then move to the SN method. Now there are two cases:

- (i) SN method terminates with a solution x^* of GLCP_r . As before, $\bar{y} = x^*/e^T x^*$ is an SP of StQP (26) with an objective function value smaller than or equal to u . This SP is given to the two-phase sequential Algorithm 4 and the enumerative algorithm terminates.
- (ii) SN method terminates with a maximum number of iterations `nmaxit` (100 by default) and Algorithm 3 continues in Step 2 with the same node k and the same point (\tilde{x}, \tilde{v}) .

From then on, we consider as the enumerative algorithm the hybrid scheme involving Algorithm 3 and the SN method in the way explained above.

6 Phase 2: giving a certificate of global optimality

Phase 1 usually terminates with an SP of StQP (26) that is a δ -approximate global minimum (and usually a global minimum) of StQP (26), where δ is given by (6). However, this phase usually fails in giving a certificate of global optimality as Algorithm 3 is, in general, unable to show that the last GLCP $_r$ has no solution. In this section, we discuss the so-called Phase 2, which gives a certificate of global optimality for the last SP computed in Phase 1.

A Nonlinear Programming Problem (NLP $_r$) is constructed based on the so-called Reformulation-Linearization Technique (RLT) [33] applied to QP $_r$ (36). This NLP $_r$ is a convex program with a compact constraint set, whence a global minimum for NLP $_r$ exists and can be computed by a local solver. Furthermore, the optimal value of this program gives a lower bound for QP $_r$. Then, one of the following cases occurs:

- (i) If NLP $_r$ is infeasible, or has a positive optimal value, then QP $_r$ is infeasible or has a positive optimal value. Then, by Theorem 7, GLCP $_r$ has no solution and a certificate of global optimality for the last computed SP computed in Phase 1 is given.
- (ii) NLP $_r$ has an optimal solution (\bar{x}, \bar{v}) with a null optimal value and satisfying $c(\bar{x}, \bar{v}) = 0$. Then $\bar{y} = \bar{x}/e^T \bar{x}$ is a new SP for StQP (26). Then we update u by (35) and apply Phase 2 again for this new SP.
- (iii) NLP $_r$ has an optimal solution (\bar{x}, \bar{v}) such that $c(\bar{x}, \bar{v}) > 0$. Then no conclusion about global optimality of the last SP computed in Phase 1 can be stated.

When the last case occurs, we propose an enumerative algorithm, that is able at least in theory to give a certificate or to find a better SP than the last SP computed in Phase 1. As before, in this case Phase 2 is applied again for this new SP. In the next subsections, we introduce the RLT convex formulation NLP $_r$ and the enumerative algorithm.

6.1 An RLT convex formulation

The RLT formulation NLP $_r$ of QP $_r$ (36) takes the following form:

$$\mathbf{NLP}_r : \min \quad g(x, v, z, \tau, t) = e^T z \quad (50a)$$

$$\text{s.t.} \quad v = Mx - e \quad (50b)$$

$$z_i = \sum_{j=1}^n m_{ij} t_{(ij)} - x_i, \quad \forall i \in N \quad (50c)$$

$$r \leq e^T x \leq s \quad (50d)$$

$$rx_i \leq \sum_{j=1}^n t_{(ij)} \leq sx_i, \quad \forall i \in N \quad (50e)$$

$$x_i^2 \leq t_{(ii)}, \quad \forall i \in N \quad (50f)$$

$$(x_i + x_j)^2 \leq t_{(ii)} + 2t_{(ij)} + t_{(jj)}, \quad \forall (i, j) \text{ with } i < j, \forall i, j \in N \quad (50g)$$

$$(x_i - x_j)^2 \leq t_{(ii)} - 2t_{(ij)} + t_{(jj)}, \quad \forall (i, j) \text{ with } i < j, \forall i, j \in N \quad (50h)$$

$$v_i^2 \leq \tau_i, \quad i \in N, \quad (50i)$$

$$(x_i + v_i)^2 \leq t_{(ii)} + 2z_i + \tau_i, \quad \forall i \in N, \quad (50j)$$

$$(x_i - v_i)^2 \leq t_{(ii)} - 2z_i + \tau_i, \quad \forall i \in N \quad (50k)$$

$$0 \leq \tau_i \leq \beta_i v_i, \quad \forall i \in N \quad (50l)$$

$$0 \leq v_i \leq \beta_i, \quad \forall i \in N \quad (50m)$$

$$x_i \geq 0, \quad \forall i \in N \quad (50n)$$

$$z_i \geq 0, \quad \forall i \in N \quad (50o)$$

$$t_{(ij)} \geq 0, \quad i \leq j, \forall i, j \in N \quad (50p)$$

where $z = (z_i) \in \mathbb{R}^n$, $\tau = (\tau_i) \in \mathbb{R}^n$, $t = (t_{ij}) \in \mathbb{R}^{n(n+1)/2}$, β_i ($i \in N$) and s are to be defined later in this section.

Next we show how to construct this NLP_r from QP_r (36) by using RLT. First, we add the following redundant constraints for $GLCP_r$:

$$(i) \quad e^T x \leq s, \quad (51)$$

where

$$s = \frac{1}{l},$$

and l is a lower bound for StQP (26) given by (27). Since $r = \frac{1}{u}$ and $u \geq l$, then (50d) holds.

$$(ii) \quad v_i \leq \beta_i, \quad \forall i \in N \quad (52)$$

where β_i is computed by

$$\begin{aligned} \beta_i = -1 + \max \quad & \sum_{j=1}^n m_{ij} x_j \\ \text{s.t.} \quad & r \leq e^T x \leq s \\ & x_i \geq 0, \quad \forall i \in N, \end{aligned}$$

where m_{ij} are the elements of the matrix M . By adding these two constraints to the definition of QP_r , we come to the conclusion that the constraint set of QP_r is compact.

The RLT technique is now used for constructing NLP_r that gives a lower bound for QP_r . To do that, the reformulation and the linearization phases are applied. These two phases are presented below.

- (i) In the reformulation phase, the following products are constructed:
- The i -th row of (36b) is multiplied by x_i , $\forall i \in N$, that is,

$$v_i x_i = \sum_{j=1}^n m_{ij} x_j x_i - x_i, \quad \forall i \in N, \quad (53)$$

where m_{ij} are the elements of the matrix M .

- The constraints $r \leq e^T x \leq s$ are multiplied by v_i , $\forall i \in N$.
- The i -th constraint (52) is multiplied by v_i , $\forall i \in N$

- (ii) In the linearization phase, the following variables are introduced to substitute the nonlinear terms as follows:

$$z_i = v_i x_i, \quad i \in N \quad (54a)$$

$$t_{(ij)} = x_i x_j = x_j x_i, \quad i, j \in N, i \leq j \quad (54b)$$

$$\tau_i = v_i^2, \quad i \in N. \quad (54c)$$

- (iii) Constraints (54b) are equivalent to:

$$\begin{aligned} x_i^2 &= t_{(ii)}, \quad \forall i \in N \\ (x_i + x_j)^2 &= t_{(ii)} + 2t_{(ij)} + t_{(jj)}, \quad \forall (i, j) \text{ with } i < j, \quad \forall i, j \in N \\ (x_i - x_j)^2 &= t_{(ii)} - 2t_{(ij)} + t_{(jj)}, \quad \forall (i, j) \text{ with } i < j, \quad \forall i, j \in N. \end{aligned}$$

Then, constraints (50f)–(50h) are convex relaxations of these three equalities.

(iv) Constraints (54a) and (54c) are equivalent to

$$\begin{aligned} v_i^2 &= \tau_i, \quad i \in N, \\ (x_i + v_i)^2 &= t_{(ii)} + 2z_i + \tau_i, \quad \forall i \in N \\ (x_i - v_i)^2 &= t_{(ii)} - 2z_i + \tau_i, \quad \forall i \in N. \end{aligned}$$

Then, constraints (50i)–(50k) are convex relaxations of these three equalities.

(v) Note also that condition (50l) implies that

$$\tau_i \leq \beta_i^2, \quad \forall i \in N. \quad (56)$$

So, the variables τ_i are bounded. Furthermore, the variables $t_{(ij)}$ and z_i are also bounded by (50e) and (50c), respectively. So, the constraint set of NLP_r is compact and NLP_r has an optimal solution provided it is feasible.

6.2 An enumerative algorithm for Phase 2

Let \bar{y} be the last SP of StQP (26) computed in Phase 1. If NLP_r (50) is infeasible, then \bar{y} is a δ -approximate global minimum of StQP (26) and Phase 2 terminates with the required certificate of global optimality. Otherwise, as the constraint set of NLP_r is compact, NLP_r (50) has an optimal solution $(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t})$. Let \bar{g} be the optimal value of NLP_r (50). If $\bar{g} > 0$, then \bar{y} is a δ -approximate global minimum of StQP (26). If $\bar{g} = 0$ and $\bar{x}^T \bar{v} = 0$, then $\bar{x}/e^T \bar{x}$ is a better SP than \bar{y} . In this case, we set $\bar{y} = \bar{x}/e^T \bar{x}$, update u by (35) and apply Phase 2 for the new SP. If $\bar{g} = 0$ and $\bar{x}^T \bar{v} > 0$, then no conclusion can be stated about \bar{y} to be a global minimum of this StQP. In this case, we apply an enumerative algorithm that is based on a branching strategy similar to the one used in Algorithm 3, that is, on $x_i = 0$ and $v_i = 0$. For each node $k \geq 1$, let I_0 and J_0 be the sets of the fixed v - and x -variables respectively, that is,

$$v_i = 0, \quad \forall i \in I_0, \quad (57a)$$

$$x_i = 0, \quad \forall i \in J_0. \quad (57b)$$

and $I_0 \cap J_0 = \emptyset$. Furthermore, let

$$K_0 = I_0 \cup J_0 \quad (58)$$

and consider the NLP_r^k given by NLP_r and the constraints:

$$z_i = 0, \quad \forall i \in K_0, \quad x_i = 0, \quad \forall i \in J_0, \quad v_i = 0, \quad \forall i \in I_0 \quad (59a)$$

$$\tau_i = 0, \quad \forall i \in I_0, \quad t_{(ij)} = 0, \quad \text{if } i \in J_0 \text{ or } j \in J_0. \quad (59b)$$

For the root node $k = 1$, we have $\text{NLP}_r^k = \text{NLP}_r$. For the remaining nodes $k > 1$, NLP_r^k is exactly NLP_r with null lower and upper bounds for all the fixed variables in (59).

In each node k , a global minimum of the convex program NLP_r^k is computed and there are three possible cases:

- (i) NLP_r^k is infeasible or has a positive optimal value. Then the node k is removed from the list of open nodes.
- (ii) NLP_r^k has an optimal solution $(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t})$ with a null optimal value and $c(\bar{x}, \bar{v}) = 0$. Then, we terminate the enumerative algorithm with a new SP $\bar{y} = \bar{x}/e^T \bar{x}$.
- (iii) NLP_r^k has an optimal solution $(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t})$ with a null optimal value and $c(\bar{x}, \bar{v}) > 0$. Then there are at least a pair of complementary variables (\bar{x}_i, \bar{v}_i) such that \bar{x}_i and \bar{v}_i are both positive. Node k is removed and two new nodes are generated from node k by fixing to zero each one of the variables of this pair. These two nodes are added to the list of open nodes.

The algorithm is finite and terminates either with a new SP of StQP (26) or with a certificate of global optimality when there are no nodes to be investigated in the list of open nodes.

The enumerative algorithm requires to solve a convex NLP at each node of the tree. A good solver for convex NLP should be used at this aim. In our experiments, we use **GUROBI** [21]. The steps of the enumerative algorithm are presented in Algorithm 4.

Algorithm 4 : Enumerative algorithm for NLP_r in Phase 2

▷ **Step 0: Initialization**

- 1: Set $\epsilon_1, \epsilon_2 > 0$ and $k = 1$.
- 2: Let I_0 and J_0 be the sets of fixed v - and x -variables (usually $I_0 = J_0 = \emptyset$) and K_0 be the set defined by (58).
- 3: Find an optimal solution $(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t})$ of NLP_r^k .
- 4: If NLP_r^k is infeasible or $g(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t}) > \epsilon_2$, terminate: $GLCP_r$ has no solution.
- 5: Otherwise, let $P = \{1\}$ be the set of open nodes and $N = 1$ be the number of generated nodes.
- 6: Set $k = 1$, $CG(1) = c(\bar{x}, \bar{v})$, where $c(x, v)$ is the objective function of QP_r (36) and $i(1) = j^*$, with j^* being the index where

$$\theta = \max\{\min\{\bar{x}_i, \bar{w}_i\} : i \in \bar{K}_0\} \quad (60)$$

is achieved.

- 7: If $c(\bar{x}, \bar{v}) \leq \epsilon_1$ OR $\theta \leq \epsilon_2$, then terminate: (\bar{x}, \bar{v}) is a solution of $GLCP_r$ and $\bar{y} = \bar{x}/e^T \bar{x}$ is an SP of StQP (26) (and of StQP (1)).

▷ **Step 1: Choice of node**

- 8: If $P = \emptyset$, terminate: $GLCP_r$ has no solution.
- 9: Otherwise, select $k \in P$ such that $CG(k) = \min\{CG(i) : i \in P\}$, set $j^* = i(k)$ and $P = P \setminus \{k\}$.
- 10: Branch on the complementary variables \bar{x}_{i^*} and \bar{v}_{i^*} associated with θ and generate two nodes $p = N + 1$ and $p = N + 2$.

▷ **Step 2: Solve, Update and Queue**

- 11: For each of $p = N + 1$ and $p = N + 2$:
 Find an optimal solution $(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t})$ of NLP_r^p defined by (50) and (59) and compute θ by (60).
 - (i) If NLP_r^p is infeasible OR $g(\bar{x}, \bar{v}, \bar{z}, \bar{\tau}, \bar{t}) > \epsilon_1$, then node p is not generated.
 - (ii) If $\theta \leq \epsilon_2$ OR $c(\bar{x}, \bar{v}) < \epsilon_1$, then terminate: (\bar{x}, \bar{v}) is a solution of $GLCP_r$ and $\bar{y} = \bar{x}/e^T \bar{x}$ is an SP of StQP (26) (and of StQP (1)).
 - (iii) Else, set $P = P \cup \{p\}$, $CG(p) = c(\bar{x}, \bar{v})$ and let $i(p)$ be the index where θ is achieved in (60).
 - 12: Return to Step 1.
-

Notes:

- (i) We set the tolerances $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-6}$ by default.
- (ii) In practice, we can use the l_∞ norm of z , that is, $\|z\|_\infty = \max\{z_i : i = 1, \dots, n\}$, instead of the l_1 norm $e^T z$ for deciding whether a node should be pruned. In this case, we set the tolerance $\epsilon_1 = 10^{-6}$ by default.

7 The two-phase sequential algorithm

In this section, we present in Algorithm 5 the steps of a two-phase sequential algorithm that exploits the ideas discussed so far.

Algorithm 5 : Two-phase sequential algorithm for StQP

▷ **Step 0: Initialization**

- 1: Let $nmaxiter_1$ and $nmaxiter_2$ be the maximum number of iterations allowed for the enumerative algorithms in Phase 1 and Phase 2 ($nitermax_1 = 100$ and $nitermax_2 = 2000$ by default). Furthermore, let α be a small positive real number ($\alpha = 10^{-3}$ by default).
 - 2: Set $Enum = 1$ (logical variable which takes the value 1 or 2 depending on the phase to be applied).
-

-
-
- 3: Compute a lower bound l_Q to StQP (1) by the procedure described in Section 3.
- 4:
- (i) Compute $f(e^r) = \min\{f(e^i) = q_{ii}, i = 1, \dots, n\}$, e^i being the i -th standard basis vector in \mathbb{R}^n and $f(y) = y^T Q y$. Compute an SP y^r of StQP (1) by using a local solver with $y^0 = e^r$ as the initial point.
 - (ii) Compute an SP y^s of StQP (1) by using a local solver with y^0 as the barycenter.
 - (iii) Compute an SP y^t of StQP (1) by using a local solver with y^0 as the vector which provides the lower bound computed in instruction 3.
 - (iv) Let $\bar{y} = \operatorname{argmin}\{f(y^r), f(y^s), f(y^t)\}$.
- 5: Let $u_Q = \bar{y}^T Q \bar{y}$. If $u_Q - l_Q \leq 0$, then \bar{y} is a global minimum of StQP (1); terminate.
- 6: **if** $l_Q \leq 0$ **then**
- 7: set $M = Q - (l_Q - \eta)E$, $l = \eta$, $u = u_Q - l_Q + \eta$, where $\eta > 0$ is a parameter to be chosen ($\eta = 1$ by default) and E is a matrix of all ones. Go to Step 1.
- 8: **else**
- 9: set $M = Q$, $l = l_Q$, and $u = u_Q$. Go to Step 1.
- 10: **end if**
- 11: Set $\bar{u} = u$.

▷ **Step 1: Update u** Compute:

$$u = \bar{u} - \alpha \max\{1, \bar{u}\}. \quad (61)$$

- 12: If $u - l \leq 0$, \bar{y} is a δ -approximate global minimum (usually a global minimum) of StQP (1), where δ is given in (6); terminate.
- 13: Let $\bar{u}_Q = \bar{y}^T Q \bar{y}$ and $u_Q = \bar{u}_Q - \alpha \max\{1, \bar{u}_Q\}$. If $u_Q - l_Q \leq 0$, \bar{y} is a δ -approximate global minimum of StQP (1), where δ is given by (6); terminate.

14: **if** Enum = 2 **then**

15: go to Step 3.

16: **end if**

▷ **Step 2: Phase 1**

17: Let $r = \frac{1}{u}$. Solve GLCP $_r$ (33) by Algorithm 3.

18: **if** Algorithm 3 terminates with a solution \bar{x} of GLCP $_r$ **then**

19: $\bar{y} = \frac{\bar{x}}{e^T \bar{x}}$ is a new SP of StQP (26) and go to Step 1 with \bar{y} and $\bar{u} = \bar{y}^T M \bar{y}$.

20: **end if**

21: **if** Algorithm 3 shows that GLCP $_r$ has no solution **then**

22: \bar{y} is a δ -approximate global minimum (usually a global minimum) for StQP (1); terminate.

23: **end if**

24: **if** Algorithm 3 stops with number of iterations $> nitermax_1$ **then**

25: set Enum = 2 and go to Step 3.

26: **end if**

▷ **Step 3: Phase 2**

27: Let $r = \frac{1}{u}$ and $s = \frac{1}{l}$. Apply Algorithm 4 for solving NLP $_r$ (50).

28: **if** Algorithm 4 terminates with a solution \bar{x} of GLCP $_r$ **then**

29: $\bar{y} = \frac{\bar{x}}{e^T \bar{x}}$ is a new SP of StQP (26) and go to Step 1 with \bar{y} and $\bar{u} = \bar{y}^T M \bar{y}$.

30: **end if**

31: **if** Algorithm 4 terminates showing that GLCP $_r$ has no solution **then**

32: \bar{y} is a δ -approximate global minimum (usually a global minimum) for StQP (1); terminate.

33: **end if**

34: **if** Algorithm 4 attains the maximum number of iterations $nitermax_2$ **then**

35: Phase 2 is unable to give a certificate of global optimality; terminate with a failure.

36: **end if**

Notes:

- (i) In Step 0, instruction 4:(i), if the same minimum value is obtained by several points $e^i, i \in \{1, 2, \dots, k\}$, $k \leq n$, then we choose $y^0 = e^r$, where r is randomly selected in the set $\{1, 2, \dots, k\}$.

- (ii) In Step 1, instruction 13:, we use the gap between the bounds of the original StQP (1) as another stopping criterion for the algorithm.
- (iii) In practice, 0 is replaced by a positive tolerance ϵ in the stopping criteria presented in instructions 5:, 12: and 13:. We set $\epsilon = 10^{-6}$ by default.

8 Computational experiments

In this section, we report the numerical experiments for solving StQPs from known sources and for establishing copositivity of some matrices that have been used in the past by several authors. In the tables to be presented in this section, we indicate the instance reference, the dimension of the matrix Q , and we use the following notation:

- l_Q and u_Q : lower and the upper bounds for StQP (1) computed in the initialization step of Algorithm 5.
- min: information about the possible termination of the algorithm before the activation of the two main phases. The possible values appearing in this column are:
 - ‘gm’ indicates that the algorithm guarantees a global minimum (or a certificate of copositivity) at Step 0, because the difference between the upper bound and the lower bound is smaller than or equal to a tolerance.
 - ‘dgm’ indicates that the algorithm guarantees at least a δ -approximate global minimum after the first computation of u in Step 1 without using the enumerative algorithm of Phase 1.
 - ‘–’ is written otherwise.
- f_{ph1} : value of the objective function of StQP (1) at the last SP computed in Phase 1. A ‘np1’ is written when Phase 1 is not required, as an SP that is at least a δ -approximate global minimum is computed in Step 0 or Step 1.
- f_{ph2} : value of the objective function of StQP (1) at the last SP computed by the enumerative algorithm in Phase 2. The other possible values appearing in this column are:
 - ‘–’ is written when no better SP was computed in Phase 2.
 - ‘np2’ indicates that Phase 2 is not required, as a certificate of global optimality is given in Phase 1.
- CallEn1: number of times that the enumerative algorithm in Phase 1 is called. A ‘–’ is written when an SP that is at least a δ -approximate global minimum is computed in Step 0 or after the first computation of u in Step 1 without using the enumerative algorithm of Phase 1.
- ItEn1: number of iterations performed in Phase 1 (0 means that only Step 0 of the enumerative algorithm is applied and – means that this algorithm is not called).
- CallEn2: number of times that the enumerative algorithm in Phase 2 is called. We use a – when Phase 2 is not required because a certificate of global optimality is given in Phase 1.
- ItEn2: number of iterations required by the enumerative algorithm in Phase 2 (0 means that only Step 0 of the enumerative algorithm is applied and – means that this algorithm is not called).
- opt: best value of the objective function of StQP (1) computed by the two-phase sequential algorithm.
- CPU: computational time in seconds. An ‘*’ following a test problem name means that the algorithm cannot terminate within the limit of CPU time allowed (7200 seconds).

The numerical experiments were performed on an Intel Core i7 clocked at 2 GHz (32 GB RAM). The StQPs required for computing the upper bounds in Step 0 and by the enumerative algorithm of Phase 1 are solved by the solver `FilterSD` [18] distributed in the `OPTI` toolbox [15]. However, we use `IPOPT` [37] for the enumerative algorithm of Phase 1 for instances concerning copositivity tests of given matrices. The convex NLPs (50) required by the enumerative algorithm of Phase 2 are solved by `GUROBI` [21]. For the test problems with dimension $n \geq 200$, the quadratic constraints (50f)-(50h) are not considered in the NLP formulation due to the limited memory of the computer used in our experiments.

As a benchmark for our analysis, we solved the same test problems by the best state-of-art B&B algorithm for StQP [27]. The code of this algorithm is available in Julia [2], and we use it by considering the best setting presented in [27], that is, a non-binary branching strategy with best-bound selection strategy. In the corresponding tables, we show the lower bound computed in the root node, the best upper bound computed

by the algorithm, the explored number of nodes, and the computational time in seconds. A ‘*’ appears when the computational time is bigger than 7200 seconds, which is set as the maximum CPU time allowed or when numerical troubles are encountered in GUROBI in Phase 2. We use the notation [gu] for this last occurrence.

8.1 Solution of StQPs from known sources

8.1.1 StQP with complex pattern

We consider StQP (1) with complex pattern where the matrix Q is generated as in [11]. Let A, B_1, \dots, B_n be symmetric matrices of order $\bar{n}, k_1, \dots, k_{\bar{n}}$, respectively. In particular, the matrix A is a symmetric randomly generated with elements uniformly distributed in the intervals $[-1, 1]$, $[-10, 10]$ and $[-100, 100]$ and dimension $n = 2, 3$, and 4, whereas B_i are identity matrices of dimension $k_1, \dots, k_{\bar{n}}$. Let us define

$$U = \begin{bmatrix} \eta_{k_1}^T & 0^T & \dots & 0^T \\ 0^T & \eta_{k_2}^T & \dots & 0^T \\ \dots & \dots & \ddots & \dots \\ 0^T & 0^T & \dots & \eta_{k_{\bar{n}}}^T \end{bmatrix},$$

where η_k is the all ones vector in \mathbb{R}^k and 0 are zero vectors of appropriate dimension. Then both $\bar{A} = U^T A U$ and $\bar{B} = \text{diag}(B_1, \dots, B_{\bar{n}})$ are symmetric matrices of order $n = \sum_{i=1}^{\bar{n}} k_i$. Finally, we generate

$$Q = \mu \bar{A} + \bar{B},$$

where μ is a large positive number. The values of $k_1, \dots, k_{\bar{n}}$ are specified in the following tables right after the collection name (e.g., the problem named Bomze20_30 has $n = 50$, $k_1 = 20$ and $k_2 = 30$).

The performance of Algorithm 5 for solving these problems is presented in Table 1. For all the test problems, the upper bound procedure in Step 0 gives a global minimum of StQP. The proposed algorithm guarantees a global minimum for nine test problems as the difference between the upper and lower computed in Step 0 is smaller than or equal to the tolerance. For the other three test problems, a δ -approximate global minimum is guaranteed without requiring the enumerative algorithm in Phase 1, as Phase 1 terminates in Step 1 after updating the upper bound. For the three instances requiring the sequential algorithm, Phase 1 cannot give a certificate of global optimality, but Phase 2 can. Only the root node is generated by the enumerative algorithm of Phase 2, that is, the optimal solution of NLP_r gives the certificate.

As shown in Table 2, the B&B algorithm computes a global optimum for all problems. For only one instance, indicated by an *, the algorithm could not terminate in the maximum CPU time allowed. We can observe that the new techniques proposed in this paper lead to very tight lower bounds which allow our algorithm to terminate with a certificate of global optimality in a short computational time. This makes Algorithm 5 to be much more efficient than B&B algorithm for the test problems of this set.

8.1.2 StQP with indefinite matrices from collections

In this section, we report the solution of StQP test problems whose matrix Q is an indefinite symmetric matrix from the Harwell-Boeing collection [28] or from the QPlib collection [19]. The performance of Algorithm 5 and the B&B algorithm for solving these problems are presented in Tables 3 and 4, respectively.

The results show that the two-phase algorithm is quite efficient for computing a global minimum and giving a certificate of global optimality for all these test problems. In particular, for all the test problems, the upper bound computed in Step 0 is a global minimum of StQP. Moreover, for one test problem, Algorithm 5 guarantees a global minimum as the difference between the upper and lower bounds computed in Step 0 is smaller than or equal to the tolerance. For two test problems, a δ -approximate global minimum is guaranteed without requiring the enumerative algorithm in Phase 1. For the other instances requiring the sequential

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	CPU
Bomze20-30	50	1.7600e+00	1.7600e+00	gm	np1	np2	-	-	-	-	1.7600e+00	2.8886e-01
Bomze25-25-25-25	100	-1.8021e+01	-3.0787e+00		-3.0787e+00	-	1	[100]	1	[0]	-3.0787e+00	1.1553e+02
Bomze50-50-100	200	-1.3866e+01	-1.3862e+01	dgm	np1	np2	-	-	-	-	-1.3862e+01	5.8300e-01
Bomze100-50-150-100	400	-3.2119e+01	-3.2118e+01	dgm	np1	np2	-	-	-	-	-3.2118e+01	2.0984e+00
Bomze200-200-200	600	-1.3877e+01	-1.3877e+01	gm	np1	np2	-	-	-	-	-1.3877e+01	5.0419e+00
Bomze20-30	50	1.7415e+01	1.7415e+01	gm	np1	np2	-	-	-	-	1.7415e+01	2.9936e-01
Bomze25-25-25-25	100	-1.8040e+02	-3.0968e+01		-3.0968e+01	-	1	[100]	1	[0]	-3.0968e+01	1.4735e+02
Bomze50-50-100	200	-1.3880e+02	-1.3880e+02	dgm	np1	np2	-	-	-	-	-1.3880e+02	7.1503e-01
Bomze100-50-150-100	400	-3.2136e+02	-3.2136e+02	gm	np1	np2	-	-	-	-	-3.2136e+02	2.1336e+00
Bomze200-200-200	600	-1.3882e+02	-1.3882e+02	gm	np1	np2	-	-	-	-	-1.3882e+02	5.3684e+00
Bomze20-30	50	1.7396e+02	1.7396e+02	gm	np1	np2	-	-	-	-	1.7396e+02	2.8901e-01
Bomze25-25-25-25	100	-1.8042e+03	-3.0986e+02		-3.0986e+02	-	1	[100]	1	[0]	-3.0986e+02	2.3920e+02
Bomze50-50-100	200	-1.3882e+03	-1.3882e+03	gm	np1	np2	-	-	-	-	-1.3882e+03	6.6804e-01
Bomze100-50-150-100	400	-3.2137e+03	-3.2138e+03	gm	np1	np2	-	-	-	-	-3.2138e+03	2.1234e+00
Bomze200-200-200	600	-1.3882e+03	-1.3882e+03	gm	np1	np2	-	-	-	-	-1.3882e+03	5.8058e+00

Table 1: Performance of Algorithm 5 for solving StQP with complex patterns.

Problem	n	l_Q	opt	Nodes	CPU
Bomze20-30	50	-7.3165e+00	1.7600e+00	335	1.0001e+01
Bomze25-25-25-25	100	-1.9743e+01	-3.0787e+00	333	2.6121e+01
Bomze50-50-100	200	-1.9320e+01	-1.3862e+01	211	8.0694e+01
Bomze100-50-150-100	400	-3.8435e+01	-3.2117e+01	199	5.6183e+02
Bomze200-200-200*	600	-2.0078e+01	-1.3877e+01	799	7.2392e+03
Bomze20-30	50	-7.3287e+01	1.7414e+01	255	9.1680e+00
Bomze25-25-25-25	100	-1.9760e+02	-3.0967e+01	337	2.8712e+01
Bomze50-50-100	200	-1.9724e+02	-1.3880e+02	266	7.3694e+01
Bomze100-50-150-100	400	-3.8863e+02	-3.2135e+02	199	5.7822e+02
Bomze200-200-200	600	-2.0185e+02	-1.3881e+02	799	6.6497e+03
Bomze20-30	50	-7.3319e+02	1.7395e+02	255	1.3446e+01
Bomze25-25-25-25	100	-1.9762e+03	-3.0985e+02	339	2.8774e+01
Bomze50-50-100	200	-1.9818e+03	-1.3882e+03	267	5.5823e+01
Bomze100-50-150-100	400	-3.8907e+03	-3.2137e+03	199	5.4191e+02
Bomze200-200-200	600	-2.0209e+03	-1.3882e+03	395	2.4357e+03

Table 2: Performance of the B&B algorithm for solving StQP with complex patterns.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	CPU
bcsprw01	39	4.6512e-02	4.7619e-02		4.7619e-02	-	1	[100]	1	[2]	4.7619e-02	9.2820e+00
bcsprw02	49	3.6364e-02	3.7037e-02	dgm	np1	np2	-	-	-	-	3.7037e-02	5.5960e-01
can...62	62	3.0769e-02	3.2258e-02		3.2258e-02	-	1	[100]	1	[0]	3.2258e-02	2.4078e+01
dwt...59	59	3.3898e-02	4.1667e-02		4.1667e-02	-	1	[100]	1	[0]	4.1667e-02	1.6330e+01
lund.a	147	1.2162e+02	1.2162e+02	dgm	np1	np2	-	-	-	-	1.2162e+02	2.0728e+01
lund.b	147	9.9422e-03	9.9422e-03	gm	np1	np2	-	-	-	-	9.9422e-03	6.8047e+00
QPLIB.0018	50	-1.8474e+01	-1.2772e+01		-1.2772e+01	-	1	[100]	1	[16]	-1.2772e+01	2.3835e+02
QPLIB.0343	50	-1.8474e+01	-1.2772e+01		-1.2772e+01	-	1	[100]	1	[0]	-1.2772e+01	1.2785e+01
QPLIB.2712	200	2.6517e-03	2.5738e-02		2.5738e-02	-	1	[100]	1	[42]	2.5738e-02	1.5003e+03

Table 3: Performance of Algorithm 5 for solving StQP with Q chosen from collections.

algorithm, Phase 1 cannot give a certificate of global optimality, but Phase 2 provides it. Few nodes are generated by the enumerative algorithm of Phase 2 to give this certificate.

The B&B algorithm computes a global minimum for all test problems, but it can give a certificate only to four of them. Moreover, a numerical error due to GUROBI in the lower bound computation is encountered for one instance. Note that the new techniques proposed in this paper lead to better lower bounds for all the problems but two. As before, Algorithm 5 is shown more efficient than the B&B algorithm for the test problems in this set.

8.1.3 StQP Nowak's problems

We consider StQPs generated by the two routines presented in [31]. In particular, we generate StQPs with known optimal value presented in [31] and used as test instances in several papers such as [32]. We set $s/n = 0.2, 0.5,$ and 0.8 where s is the number of negative eigenvalues of the Hessian of $f(y)$, and we choose the number of non-binding constraints k at the optimal solution equal to $0.1n$. We also generate StQPs with known density of the convexity graphs. We set the density equal to $0.25, 0.5,$ and 0.75 as in [32] and [27] and we choose $n = 100$.

Problem	n	l_Q	opt	Nodes	CPU
bcspr01	39	0.0000e+00	4.7619e-02	123789	2.5743e+03
bcspr02*	49	0.0000e+00	3.7037e-02	333979	7.2000e+03
can...62*	62	0.0000e+00	3.1249e-02	153983	7.2001e+03
dwt...59*	59	0.0000e+00	4.1666e-02	27889	7.2012e+03
lund.a*	147	-6.4998e+07	1.2160e+02	3902 [gu]	3.0025e+03
lund.b*	147	-1.6414e+03	9.9420e-03	9625	7.2020e+03
QPLIB.0018	50	-1.4813e+01	-1.2772e+01	104	2.8565e+01
QPLIB.0343	50	-1.4813e+01	-1.2772e+01	104	7.0434e+01
QPLIB.2712	200	-2.8074e-01	2.5736e-02	81	1.1581e+02

Table 4: Performance of the B&B algorithm for solving StQP with Q chosen from collections.

The results for the first set of test problems obtained by Algorithm 5 are presented in Table 5 and show that the algorithm easily finds a global minimum and easily gives a certificate of global optimality for all the test problems. For all the test problems, the upper bound procedure in Step 0 gives a global minimum of StQP. A δ -approximate global minimum is guaranteed without requiring the enumerative algorithm of Phase 1 for six test problems. For the remaining test problems, the enumerative algorithm in Phase 1 can give a certificate of global optimality, and Phase 2 is not required. The last column shows the accuracy of the solution computed by Algorithm 5. In particular, it reports the maximum relative error, that is, $\%MRE = 100 \frac{f(\bar{y}) - Opt}{Opt}$, where Opt is the optimal global value given by the collection, and \bar{y} is the δ -approximate global optimal solution computed by one of the algorithms. The values of $\%MRE$ indicate that Algorithm 5 was able to calculate an accurate global minimum for each test problem. Moreover, the performance of Algorithm 5 seems to be comparable to a special purpose heuristic algorithm discussed in [32] for all the test problems of this set. Finally, we mention here that B&B algorithm is unable to solve all these test problems for numerical troubles in the initialization phase.

The numerical results corresponding to the second set of test problems solved by Algorithm 5 are given in Table 6. We observed that the computational effort required by GUROBI to solve NLP (50) is quite high in this case. Hence, we excluded the constraints (50g)-(50h). Algorithm 5 can compute a global solution and give a certificate for all problems but one. A global minimum was computed in Phase 1 for all instances, but one. For this instance, Phase 2 was able to improve the optimal value and give the certificate of global optimality. Table 7 indicates that the B&B algorithm has a very good performance for the test problems in this set both in terms of computational time and in the computed lower bounds.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	CPU	%MRE
$s/n = 0.2, k/n = 0.1$													
Nowak-30	30	-2.2012e+05	-2.2010e+05	dgm	np1	np2	-	-	-	-	-2.2010e+05	3.8439e-01	3.6000e-05
Nowak-50	50	-2.5319e+05	-2.5314e+05	dgm	np1	np2	-	-	-	-	-2.5314e+05	7.9577e-01	6.0000e-05
Nowak-100	100	-1.2120e+05	-1.2039e+05		-1.2039e+05	np2	1	[0]	-	-	-1.2039e+05	8.1662e+00	1.1993e-04
Nowak-200	200	-2.8489e+05	-2.8362e+05		-2.8362e+05	np2	1	[0]	-	-	-2.8362e+05	4.0712e+00	2.4000e-04
Nowak-250	250	-4.1050e+05	-4.0884e+05		-4.0884e+05	np2	1	[0]	-	-	-4.0884e+05	8.3094e+00	2.9999e-04
$s/n = 0.5, k/n = 0.1$													
Nowak-30	30	-2.2012e+05	-2.2010e+05	dgm	np1	np2	-	-	-	-	-2.2010e+05	1.2317e-01	4.6000e-05
Nowak-50	50	-2.5315e+05	-2.5314e+05	dgm	np1	np2	-	-	-	-	-2.5314e+05	2.7950e-01	7.6000e-05
Nowak-100	100	-1.2112e+05	-1.2039e+05		-1.2039e+05	np2	1	[0]	-	-	-1.2039e+05	2.8861e+00	1.5000e-04
Nowak-200	200	-2.8439e+05	-2.8362e+05		-2.8362e+05	np2	1	[0]	-	-	-2.8362e+05	3.6474e+00	3.0000e-04
Nowak-250	250	-4.1008e+05	-4.0884e+05		-4.0884e+05	np2	1	[0]	-	-	-4.0884e+05	2.0717e+00	3.7600e-04
$s/n = 0.8, k/n = 0.1$													
Nowak-30	30	-2.2010e+05	-2.2010e+05	dgm	np1	np2	-	-	-	-	-2.2010e+05	8.6664e-02	5.4000e-05
Nowak-50	50	-2.5314e+05	-2.5314e+05	dgm	np1	np2	-	-	-	-	-2.5314e+05	1.2023e-01	9.0000e-05
Nowak-100	100	-1.2061e+05	-1.2039e+05		-1.2061e+05	np2	1	[0]	-	-	-1.2039e+05	2.0521e+00	1.8000e-04
Nowak-200	200	-2.8400e+05	-2.8362e+05		-2.8362e+05	np2	1	[0]	-	-	-2.8362e+05	1.2127e+00	3.6000e-04
Nowak-250	250	-4.0945e+05	-4.0884e+05		-4.0884e+05	np2	1	[0]	-	-	-4.0884e+05	2.5126e+00	4.5000e-04

Table 5: Performance of Algorithm 5 for solving StQP Nowak's problems with known optimal solution.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	CPU
Problem(0.25)	100	-9.4108e+00	-5.4743e+00	-5.4743e+00	-	-	1	[200]	1	[474]	-5.4743e+00	7.8767e+02
Problem(0.25) ₁	100	-9.3889e+00	-5.8544e+00	-5.8544e+00	-	-	1	[200]	1	[392]	-5.8544e+00	7.0618e+02
Problem(0.25) ₂	100	-9.4505e+00	-5.2993e+00	-5.5553e+00	-	-	4	[29 29 99 200]	1	[498]	-5.5553e+00	8.8649e+02
Problem(0.25) ₃	100	-9.4145e+00	-5.3129e+00	-5.6151e+00	-	-	2	[35 200]	1	[442]	-5.6151e+00	7.6792e+02
Problem(0.25) ₄	100	-9.4731e+00	-5.4722e+00	-5.6801e+00	-5.6856e+00	-	2	[2 200]	2	[270 500]	-5.6856e+00	1.4973e+03
Problem(0.25) ₅	100	-9.4480e+00	-6.0688e+00	-6.0688e+00	-	-	1	[200]	1	[304]	-6.0688e+00	5.5442e+02
Problem(0.5)	100	-9.5458e+00	-6.0815e+00	-6.2514e+00	-	-	4	[106 194 137 200]	1	[1300]	-6.2514e+00	2.4098e+03
Problem(0.5) ₁	100	-9.4119e+00	-6.0793e+00	-6.0793e+00	-	-	1	[200]	1	[1346]	-6.0793e+00	2.4267e+03
Problem(0.5) ₂	100	-9.5519e+00	-5.7899e+00	-6.2773e+00	-	-	3	[1 129 200]	1	[1252]	-6.2773e+00	2.6472e+03
Problem(0.5) ₃	100	-9.5853e+00	-5.9070e+00	-6.0745e+00	-	-	2	[81 200]	1	[1858]	-6.0745e+00	3.6210e+03
Problem(0.5) ₄	100	-9.4666e+00	-5.9700e+00	-6.2675e+00	-	-	3	[5 0 200]	1	[950]	-6.2675e+00	1.8655e+03
Problem(0.5) ₅	100	-9.5706e+00	-5.7478e+00	-6.1810e+00	-	-	3	[6 19 200]	1	[1536]	-6.1810e+00	2.5471e+03
Problem(0.75)	100	-9.5459e+00	-6.6027e+00	-6.6027e+00	-	-	1	[200]	1	[3138]	-6.6027e+00	5.4217e+03
Problem(0.75) ₁	100	-9.4121e+00	-6.4670e+00	-6.5740e+00	-	-	3	[178 22 200]	1	[2258]	-6.5740e+00	3.9578e+03
Problem(0.75) ₂ *	100	-9.5520e+00	-6.2129e+00	-6.3892e+00	nocert	-	3	[0 11 200]	1	[4215]	-6.3892e+00	7.2020e+03
Problem(0.75) ₃	100	-9.5854e+00	-6.6184e+00	-6.6184e+00	-	-	1	[200]	1	[2918]	-6.6184e+00	4.7676e+03
Problem(0.75) ₄	100	-9.4669e+00	-6.2836e+00	-6.5648e+00	-	-	3	[27 7 200]	1	[2262]	-6.5648e+00	3.9174e+03
Problem(0.75) ₅	100	-9.5706e+00	-6.6382e+00	-6.8131e+00	-	-	4	[3 100 64 200]	1	[2130]	-6.8131e+00	3.7538e+03
Problem(0.9)	100	-9.4122e+00	-6.7290e+00	-7.0148e+00	-	-	2	[153 200]	1	[1866]	-7.0148e+00	3.5066e+03
Problem(0.9) ₁	100	-9.5524e+00	-6.9327e+00	-6.9327e+00	-	-	1	[200]	1	[2776]	-6.9327e+00	4.8708e+03
Problem(0.9) ₂	100	-9.5854e+00	-6.8662e+00	-6.8662e+00	-	-	1	[200]	1	[3490]	-6.8662e+00	6.0075e+03
Problem(0.9) ₃	100	-9.4669e+00	-6.7313e+00	-6.7313e+00	-	-	1	[200]	1	[3426]	-6.7313e+00	6.0854e+03
Problem(0.9) ₄	100	-9.5705e+00	-7.1489e+00	-7.1489e+00	-	-	1	[200]	1	[2184]	-7.1489e+00	4.0196e+03
Problem(0.9) ₅	100	-9.4683e+00	-6.4806e+00	-6.7201e+00	-	-	3	[1 24 200]	1	[4202]	-6.7201e+00	7.1497e+03

Table 6: Performance of Algorithm 5 for solving StQP Nowak's problems with known density of the convexity graph.

Problem	n	l_Q	opt	Nodes	CPU
Problem(0.25)	100	-7.1500e+00	-5.4743e+00	201	7.1330e+00
Problem(0.25) ₁	100	-7.1553e+00	-5.8544e+00	128	3.6700e+00
Problem(0.25) ₂	100	-7.0667e+00	-5.5553e+00	190	4.5010e+00
Problem(0.25) ₃	100	-6.9246e+00	-5.6151e+00	138	3.6180e+00
Problem(0.25) ₄	100	-7.2308e+00	-5.6885e+00	174	3.9530e+00
Problem(0.25) ₅	100	-7.1861e+00	-6.0688e+00	89	2.6230e+00
Problem(0.5)	100	-7.6569e+00	-6.2799e+00	373	1.5282e+01
Problem(0.5) ₁	100	-7.5291e+00	-6.1407e+00	319	1.3204e+01
Problem(0.5) ₂	100	-7.5743e+00	-6.2773e+00	348	1.4833e+01
Problem(0.5) ₃	100	-7.6626e+00	-6.2793e+00	343	1.4427e+01
Problem(0.5) ₄	100	-7.5391e+00	-6.2675e+00	218	1.0259e+01
Problem(0.5) ₅	100	-7.6700e+00	-6.1810e+00	454	1.8281e+01
Problem(0.75)	100	-7.9236e+00	-6.6304e+00	884	7.8136e+01
Problem(0.75) ₁	100	-7.8189e+00	-6.5740e+00	578	5.4983e+01
Problem(0.75) ₂	100	-7.8316e+00	-6.5390e+00	867	8.1292e+01
Problem(0.75) ₃	100	-7.9395e+00	-6.6311e+00	697	6.5352e+01
Problem(0.75) ₄	100	-7.8225e+00	-6.5648e+00	584	5.8637e+01
Problem(0.75) ₅	100	-7.8890e+00	-6.8131e+00	381	4.3505e+01
Problem(0.9)	100	-7.9664e+00	-7.0148e+00	317	5.5809e+01
Problem(0.9) ₁	100	-7.9537e+00	-6.9327e+00	619	1.0117e+02
Problem(0.9) ₂	100	-8.0346e+00	-6.8661e+00	882	1.4635e+02
Problem(0.9) ₃	100	-7.9242e+00	-6.7313e+00	922	1.5392e+02
Problem(0.9) ₄	100	-8.0069e+00	-7.1489e+00	315	6.4388e+01
Problem(0.9) ₅	100	-7.9429e+00	-6.7201e+00	1386	2.3096e+02

Table 7: Performance of the B&B algorithm for solving StQP Nowak's problems with known density of the convexity graph.

8.1.4 StQP BLST problems

In this section, we report the performance of the two-phase sequential and the B&B algorithms on the test problems introduced in [12] and [20]. These test problems consist of random matrices generated from a triangular distribution characterized by the following parameters: minimum value a , maximum value b and mode c . These values are indicated in the title of each instance. Moreover, we set all the diagonal elements of the matrix Q of the StQP to be nonnegative.

In Tables 8 and 9, we report the solution by Algorithm 5 and the B&B algorithm for this set of test problems when the density of the matrix Q is equal to 0.5. For all instances, the two-phase sequential algorithm computes a global minimum in Phase 1 and Phase 2 gives a certificate of global optimality. In general, few nodes are generated by the enumerative algorithm in Phase 2. For ten instances, the upper bound computed in the initialization step is the global optimum value. For the remaining instances, the enumerative

algorithm of Phase 1 was called, in general, more than once in order to compute a global minimum. For these instances, the B&B algorithm shows to have a similar performance in terms of CPU time.

We performed additional numerical experiments, not reported in this paper, by considering the density of the matrix Q to be equal to 0.75 and 1 (full dense matrix). In the first case, the two algorithms showed a similar performance, and successfully computed a global minimum and provide a certificate of global optimality for all the instances. On the other hand, for the full dense matrices, Algorithm 5 was able to solve all the instances, while the B&B algorithm fails on ten instances.

As a final conclusion of the solution of all these StQP test problems, we claim that the two-phase sequential algorithm is, in general, efficient for computing a global minimum by a finding a sequence of SPs with strictly decreasing function values. The (hybrid) enumerative algorithm of Phase 1 performs well for all these instances with no branching or a small number of nodes. When Phase 1 is not able to compute a better SP, the enumerative algorithm attains the limit number of 100 nodes and Phase 2 comes to operation. This explains the occurrence of the number 100 for the last GLCP_r processed in Phase 1 when Phase 2 is required. The new techniques for computing lower bounds usually lead to tight values. This allowed the algorithm to terminate in Phase 1 in many cases. Phase 2 has shown to be efficient to give a certificate of global optimality for the last SP computed in Phase 1, when such an SP is, in fact, a global minimum. Otherwise, Phase 2 may struggle, as it has difficulties to compute a better SP. This is not surprising, as Phase 2 is not designed to compute an SP, but instead to give a certificate of global optimality for an SP. This confirms our claim to have two separate phases for dealing with global optimization problems. Furthermore, the algorithm seems to be competitive with and in many cases superior to the best state-of-art B&B algorithm and a special-purpose heuristic algorithm that has been proposed for the solution of Nowak's test problems.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	CPU
BSLT.tri.-10.0.-5.0.5.pos.1	50	-8.5085e+00	-4.2656e+00	-4.4259e+00	-	-	2	[43 100]	1	[12]	-4.4259e+00	1.8212e+02
BSLT.tri.-10.0.-5.0.5.pos.2	50	-8.1998e+00	-3.9555e+00	-4.0061e+00	-	-	3	[0 1 100]	1	[18]	-4.0061e+00	2.3401e+02
BSLT.tri.-10.0.-5.0.5.pos.3	50	-8.3982e+00	-4.5122e+00	-4.5122e+00	-	-	1	[100]	1	[2]	-4.5122e+00	3.4932e+01
BSLT.tri.-10.0.-5.0.5.pos.4	50	-8.3105e+00	-4.0147e+00	-4.0147e+00	-	-	1	[100]	1	[0]	-4.0147e+00	1.8136e+01
BSLT.tri.-10.0.-5.0.5.pos.5	50	-8.2830e+00	-4.3635e+00	-4.4868e+00	-	-	3	[3 41 100]	1	[6]	-4.4868e+00	8.5029e+01
BSLT.tri.-10.0.-5.0.5.pos.6	50	-8.0470e+00	-4.2651e+00	-4.3528e+00	-	-	2	[1 100]	1	[12]	-4.3528e+00	1.6591e+02
BSLT.tri.-10.0.-5.0.5.pos.7	50	-8.7647e+00	-4.0004e+00	-4.1372e+00	-	-	2	[0 100]	1	[0]	-4.1372e+00	1.4046e+01
BSLT.tri.-10.0.-5.0.5.pos.8	50	-8.2495e+00	-4.0666e+00	-4.0666e+00	-	-	1	[100]	1	[10]	-4.0666e+00	1.4062e+02
BSLT.tri.-10.0.-5.0.5.pos.9	50	-7.7966e+00	-4.0373e+00	-4.0373e+00	-	-	1	[100]	1	[22]	-4.0373e+00	2.8114e+02
BSLT.tri.-10.0.-5.0.5.pos.10	50	-8.6071e+00	-4.6547e+00	-4.6547e+00	-	-	1	[100]	1	[6]	-4.6547e+00	1.0505e+02
BSLT.tri.-10.10.0.0.5.pos.1	50	-7.6226e+00	-2.7461e+00	-3.1713e+00	-	-	3	[1 3 100]	1	[0]	-3.1713e+00	1.5240e+01
BSLT.tri.-10.10.0.0.5.pos.2	50	-7.0798e+00	-2.6494e+00	-2.8420e+00	-	-	2	[17 100]	1	[0]	-2.8420e+00	1.4776e+01
BSLT.tri.-10.10.0.0.5.pos.3	50	-7.4269e+00	-2.1799e+00	-2.9611e+00	-	-	3	[15 3 100]	1	[2]	-2.9611e+00	4.7019e+01
BSLT.tri.-10.10.0.0.5.pos.4	50	-7.3634e+00	-2.4781e+00	-2.8376e+00	-	-	5	[5 21 33 11 100]	1	[2]	-2.8376e+00	4.1805e+01
BSLT.tri.-10.10.0.0.5.pos.5	50	-7.2315e+00	-3.3573e+00	-3.3840e+00	-	-	2	[0 100]	1	[0]	-3.3840e+00	1.7237e+01
BSLT.tri.-10.10.0.0.5.pos.6	50	-6.7586e+00	-2.7076e+00	-3.1663e+00	-	-	2	[33 100]	1	[0]	-3.1663e+00	1.5184e+01
BSLT.tri.-10.10.0.0.5.pos.7	50	-8.1759e+00	-3.4282e+00	-3.4282e+00	-	-	1	[100]	1	[0]	-3.4282e+00	2.0146e+01
BSLT.tri.-10.10.0.0.5.pos.8	50	-7.1920e+00	-2.9463e+00	-2.9463e+00	-	-	1	[100]	1	[0]	-2.9463e+00	1.6082e+01
BSLT.tri.-10.10.0.0.5.pos.9	50	-6.2954e+00	-2.8618e+00	-3.1471e+00	-	-	2	[47 100]	1	[4]	-3.1471e+00	8.6013e+01
BSLT.tri.-10.10.0.0.5.pos.10	50	-7.8872e+00	-2.7692e+00	-2.7934e+00	-	-	2	[1 100]	1	[0]	-2.7934e+00	1.6197e+01
BSLT.tri.-10.10.3.0.5.pos.1	50	-7.3438e+00	-1.9440e+00	-2.8516e+00	-	-	4	[9 27 47 100]	1	[0]	-2.8516e+00	1.7279e+01
BSLT.tri.-10.10.3.0.5.pos.2	50	-6.7343e+00	-2.5291e+00	-2.5985e+00	-	-	2	[9 100]	1	[0]	-2.5985e+00	1.4930e+01
BSLT.tri.-10.10.3.0.5.pos.3	50	-7.1412e+00	-1.8034e+00	-2.6586e+00	-	-	3	[1 35 100]	1	[0]	-2.6586e+00	1.8562e+01
BSLT.tri.-10.10.3.0.5.pos.4	50	-7.0520e+00	-1.8630e+00	-2.8524e+00	-	-	5	[3 43 34 19 100]	1	[0]	-2.8524e+00	1.7799e+01
BSLT.tri.-10.10.3.0.5.pos.5	50	-6.8932e+00	-2.8298e+00	-2.9833e+00	-	-	2	[11 100]	1	[0]	-2.9833e+00	1.7645e+01
BSLT.tri.-10.10.3.0.5.pos.6	50	-6.3678e+00	-2.3127e+00	-2.6482e+00	-	-	2	[25 100]	1	[4]	-2.6482e+00	8.6206e+01
BSLT.tri.-10.10.3.0.5.pos.7	50	-7.9756e+00	-2.9443e+00	-2.9607e+00	-	-	2	[3 100]	1	[0]	-2.9607e+00	1.6216e+01
BSLT.tri.-10.10.3.0.5.pos.8	50	-6.8768e+00	-2.2799e+00	-2.6085e+00	-	-	2	[0 100]	1	[2]	-2.6085e+00	4.1576e+01
BSLT.tri.-10.10.3.0.5.pos.9	50	-5.8343e+00	-2.8371e+00	-2.8371e+00	-	-	1	[100]	1	[0]	-2.8371e+00	1.8506e+01
BSLT.tri.-10.10.3.0.5.pos.10	50	-7.6444e+00	-1.9770e+00	-1.9980e+00	-	-	2	[7 100]	1	[2]	-1.9980e+00	4.7498e+01
BSLT.tri.-10.10.-3.0.5.pos.1	50	-7.9202e+00	-3.5612e+00	-3.5612e+00	-	-	1	[100]	1	[0]	-3.5612e+00	1.5705e+01
BSLT.tri.-10.10.-3.0.5.pos.2	50	-7.4659e+00	-2.9786e+00	-3.0892e+00	-	-	2	[89 100]	1	[2]	-3.0892e+00	3.8080e+01
BSLT.tri.-10.10.-3.0.5.pos.3	50	-7.7488e+00	-2.6736e+00	-2.8318e+00	-	-	4	[3 5 11 100]	1	[4]	-2.8318e+00	6.5798e+01
BSLT.tri.-10.10.-3.0.5.pos.4	50	-7.6926e+00	-3.0085e+00	-3.0654e+00	-	-	2	[0 100]	1	[0]	-3.0654e+00	1.6276e+01
BSLT.tri.-10.10.-3.0.5.pos.5	50	-7.5896e+00	-3.7874e+00	-3.7874e+00	-	-	1	[100]	1	[4]	-3.7874e+00	6.9104e+01
BSLT.tri.-10.10.-3.0.5.pos.6	50	-7.2036e+00	-3.0431e+00	-3.5962e+00	-	-	3	[1 21 100]	1	[0]	-3.5962e+00	1.5255e+01
BSLT.tri.-10.10.-3.0.5.pos.7	50	-8.3825e+00	-2.9327e+00	-3.6633e+00	-	-	2	[1 100]	1	[0]	-3.6633e+00	2.0998e+01
BSLT.tri.-10.10.-3.0.5.pos.8	50	-7.5364e+00	-2.8684e+00	-3.0397e+00	-	-	2	[23 100]	1	[4]	-3.0397e+00	8.5044e+01
BSLT.tri.-10.10.-3.0.5.pos.9	50	-6.7908e+00	-3.0606e+00	-3.3740e+00	-	-	3	[15 61 100]	1	[2]	-3.3740e+00	6.0388e+01
BSLT.tri.-10.10.-3.0.5.pos.10	50	-8.1545e+00	-3.3386e+00	-3.3425e+00	-	-	2	[26 100]	1	[0]	-3.3425e+00	1.6003e+01

Table 8: Performance of Algorithm 5 for solving StQP BLST problems (density = 0.5).

Problem	n	l_Q	opt	Nodes	CPU
BSLT_tri_-10.0_-5.0.50_pos.1	50	-5.8924e+00	-4.4259e+00	427	1.7036e+01
BSLT_tri_-10.0_-5.0.50_pos.2	50	-5.8535e+00	-4.2548e+00	629	3.0387e+01
BSLT_tri_-10.0_-5.0.50_pos.3	50	-5.8848e+00	-4.5122e+00	227	1.3754e+01
BSLT_tri_-10.0_-5.0.50_pos.4	50	-5.5622e+00	-4.0147e+00	1215	4.6621e+01
BSLT_tri_-10.0_-5.0.50_pos.5	50	-6.0958e+00	-4.4868e+00	823	3.2455e+01
BSLT_tri_-10.0_-5.0.50_pos.6	50	-5.8460e+00	-4.3528e+00	665	2.6039e+01
BSLT_tri_-10.0_-5.0.50_pos.7	50	-5.8188e+00	-4.1372e+00	1069	4.1723e+01
BSLT_tri_-10.0_-5.0.50_pos.8	50	-5.9052e+00	-4.0666e+00	1943	8.0810e+01
BSLT_tri_-10.0_-5.0.50_pos.9	50	-5.7516e+00	-4.0373e+00	2871	1.1146e+02
BSLT_tri_-10.0_-5.0.50_pos.10	50	-5.9550e+00	-4.6547e+00	235	1.0341e+01
BSLT_tri_-10.10.0.0.50_pos.1	50	-4.1227e+00	-3.1714e+00	33	2.2320e+00
BSLT_tri_-10.10.0.0.50_pos.2	50	-3.5692e+00	-2.8420e+00	59	2.9900e+00
BSLT_tri_-10.10.0.0.50_pos.3	50	-3.8532e+00	-2.9611e+00	45	2.4040e+00
BSLT_tri_-10.10.0.0.50_pos.4	50	-3.6324e+00	-2.8376e+00	43	2.0260e+00
BSLT_tri_-10.10.0.0.50_pos.5	50	-4.0926e+00	-3.3840e+00	56	2.4930e+00
BSLT_tri_-10.10.0.0.50_pos.6	50	-3.8729e+00	-3.1663e+00	32	2.0710e+00
BSLT_tri_-10.10.0.0.50_pos.7	50	-3.7651e+00	-3.4282e+00	27	1.8460e+00
BSLT_tri_-10.10.0.0.50_pos.8	50	-3.6987e+00	-2.9463e+00	33	1.9880e+00
BSLT_tri_-10.10.0.0.50_pos.9	50	-3.5891e+00	-3.1471e+00	20	1.4900e+00
BSLT_tri_-10.10.0.0.50_pos.10	50	-3.9576e+00	-2.7934e+00	81	3.5570e+00
BSLT_tri_-10.10.3.0.50_pos.1	50	-3.7011e+00	-2.8516e+00	50	2.3370e+00
BSLT_tri_-10.10.3.0.50_pos.2	50	-3.1781e+00	-2.5985e+00	26	1.9060e+00
BSLT_tri_-10.10.3.0.50_pos.3	50	-3.2674e+00	-2.6586e+00	25	1.6810e+00
BSLT_tri_-10.10.3.0.50_pos.4	50	-3.1202e+00	-2.8524e+00	16	1.2010e+00
BSLT_tri_-10.10.3.0.50_pos.5	50	-3.6767e+00	-2.9833e+00	41	2.0360e+00
BSLT_tri_-10.10.3.0.50_pos.6	50	-3.2711e+00	-2.6482e+00	26	1.5410e+00
BSLT_tri_-10.10.3.0.50_pos.7	50	-3.2148e+00	-2.9607e+00	19	1.4390e+00
BSLT_tri_-10.10.3.0.50_pos.8	50	-3.3619e+00	-2.6085e+00	45	2.1240e+00
BSLT_tri_-10.10.3.0.50_pos.9	50	-3.1374e+00	-2.8371e+00	13	1.1840e+00
BSLT_tri_-10.10.3.0.50_pos.10	50	-3.2306e+00	-2.2663e+00	60	2.9390e+00
BSLT_tri_-10.10.-3.0.50_pos.1	50	-4.5201e+00	-3.5612e+00	46	2.5210e+00
BSLT_tri_-10.10.-3.0.50_pos.2	50	-4.1886e+00	-3.0892e+00	151	5.6150e+00
BSLT_tri_-10.10.-3.0.50_pos.3	50	-4.3327e+00	-3.6029e+00	32	2.1020e+00
BSLT_tri_-10.10.-3.0.50_pos.4	50	-4.1512e+00	-3.0654e+00	90	3.6510e+00
BSLT_tri_-10.10.-3.0.50_pos.5	50	-4.6555e+00	-3.7874e+00	51	2.5780e+00
BSLT_tri_-10.10.-3.0.50_pos.6	50	-4.4631e+00	-3.5962e+00	39	2.0780e+00
BSLT_tri_-10.10.-3.0.50_pos.7	50	-4.2149e+00	-3.6633e+00	29	1.9540e+00
BSLT_tri_-10.10.-3.0.50_pos.8	50	-4.1511e+00	-3.0397e+00	79	3.8270e+00
BSLT_tri_-10.10.-3.0.50_pos.9	50	-4.0899e+00	-3.3740e+00	33	1.8250e+00
BSLT_tri_-10.10.-3.0.50_pos.10	50	-4.6101e+00	-3.3425e+00	98	4.2090e+00

Table 9: Performance of the B&B algorithm for solving StQP BLST problems (density = 0.5).

8.2 Verifying copositivity of matrices from collections

Algorithm 5 can be adapted to verify whether Q is a copositive matrix, i.e., verify if $Q \in \mathcal{COP}$. In particular, the algorithm terminates with a conclusion about copositivity of the matrix Q in the two situations mentioned in the following property:

Property 1 Let l_Q and u_Q be a lower bound and an upper bound for StQP (1).

- (i) If $l_Q \geq 0$, then $Q \in \mathcal{COP}$.
- (ii) If $u_Q < 0$, then $Q \notin \mathcal{COP}$.

So, in Step 0 of Algorithm 5, we either have a certificate of copositivity or

$$l_Q < 0 \leq u_Q. \quad (62)$$

Property 2 Let η and u be given in Step 0 and Step 1 of Algorithm 5. If $u - \eta \leq 0$, then $Q \in \mathcal{COP}$.

So, Algorithm 5 terminates in Step 1, after updating the upper bound u by (61) if this property holds.

Property 3 (i) Assume that GLCP_r has a solution \bar{x} , and let $\bar{y} = \frac{\bar{x}}{e^T \bar{x}}$, $\bar{u} = \bar{y}^T M \bar{y}$ and let l_Q be a lower bound for StQP (1). If $\bar{u} - \eta + l_Q < 0$, then $Q \notin \mathcal{COP}$.

(ii) If a GLCP_r has no solution, then $Q \in \mathcal{COP}$.

Notes:

- (i) In practice, 0 is replaced by a positive tolerance ϵ for verifying these properties. We use $\epsilon = 10^{-6}$ by default.
- (ii) We implemented the conditions of these three properties in the B&B algorithm that we use to compare with Algorithm 5.

In our experiments, we start by investigating the copositivity of seven matrices that have been introduced in some papers on this subject. These matrices are presented below.

(a) Noncopositive matrices [13, 26]:

$$Q_1 = \begin{bmatrix} 1 & -0.72 & -0.59 & 1 \\ -0.72 & 1 & -0.6 & -0.46 \\ -0.59 & -0.6 & 1 & -0.6 \\ 1 & -0.46 & -0.6 & 1 \end{bmatrix} \text{ and } Q_2 = \begin{bmatrix} 1 & -0.72 & -0.59 & -0.6 \\ -0.72 & 1 & 0.21 & -0.46 \\ -0.59 & 0.21 & 1 & -0.6 \\ -0.6 & -0.46 & -0.6 & 1 \end{bmatrix}$$

(b) Strictly copositive matrices [26]:

$$Q_3 = \begin{bmatrix} 1 & 0.9 & -0.54 & 0.21 \\ 0.9 & 1 & -0.03 & 0.78 \\ -0.54 & -0.03 & 1 & 0.52 \\ 0.21 & 0.78 & 0.52 & 1 \end{bmatrix} \text{ and } Q_4 = \begin{bmatrix} 1 & 0.9 & -0.54 \\ 0.9 & 1 & -0.03 \\ -0.54 & -0.03 & 1 \end{bmatrix}$$

(c) Copositive matrices that are not strictly copositive matrices [22, 23, 36]:

$$Q_5 = \begin{bmatrix} 1 & -1 & 1 & 2 & -3 \\ -1 & 2 & -3 & -3 & 4 \\ 1 & -3 & 5 & 6 & -4 \\ 2 & -3 & 6 & 5 & -8 \\ -3 & 4 & -4 & -8 & 16 \end{bmatrix}, \quad Q_6 = \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}$$

and

$$Q_7 = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 1 & -1 \\ -1 & 1 & -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 & -1 & 1 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}$$

For all the test problems, a conclusion about the copositivity of the corresponding matrices was established. For the first two instances, Algorithm 5 terminates in Step 0 as the computed upper bound is negative. Then the matrices are declared to be not copositive. The algorithm guarantees a global minimum for the two test problems in (b) as the difference between the upper and the lower bounds computed in Step 0 is smaller than or equal to the tolerance. Furthermore, since the computed bounds are nonnegative, then the matrices are declared to be copositive. For the last three test problems, the enumerative algorithm in Phase 1 can give a certificate of global optimality, and Phase 2 is not required. Since the global optimal value is zero, the matrices are copositive.

Table 11 shows that also the B&B algorithm correctly verifies the copositivity of the matrices. The performance of the two algorithms is comparable for these instances.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	test	CPU
Q_1	4	-9.4836e-02	-9.1859e-02	-	np1	np2	-	-	-	-	-9.1859e-02	$Q \notin \mathcal{COP}$	2.4099e-01
Q_2	4	-1.1638e-01	-1.1638e-01	-	np1	np2	-	-	-	-	-1.1638e-01	$Q \notin \mathcal{COP}$	2.5181e-01
Q_3	4	2.3000e-01	2.3000e-01	gm	np1	np2	-	-	-	-	2.3000e-01	$Q \in \mathcal{COP}$	2.6005e-01
Q_4	3	2.3000e-01	2.3000e-01	gm	np1	np2	-	-	-	-	2.3000e-01	$Q \in \mathcal{COP}$	2.4135e-01
Q_5	5	-3.0476e+00	5.2979e-16		5.2979e-16	np2	1	[20]	-	-	5.2979e-16	$Q \in \mathcal{COP}$	9.1113e-01
Q_6	5	-2.0000e-01	1.0526e-16		1.0526e-16	np2	1	[20]	-	-	1.0526e-16	$Q \in \mathcal{COP}$	1.7594e+00
Q_7	7	-4.2857e-01	0.0000e+00		0.0000e+00	np2	1	[28]	-	-	0.0000e+00	$Q \in \mathcal{COP}$	2.2952e+00

Table 10: Performance of Algorithm 5 for testing copositivity of Q_1 - Q_7 .

Problem	n	l_Q	opt	Nodes	test	CPU
Q_1	4	-5.9276e-01	-9.1859e-02	0	$Q \notin \mathcal{COP}$	5.5050e+00
Q_2	4	-5.9532e-01	-1.1638e-01	0	$Q \notin \mathcal{COP}$	1.6000e+00
Q_3	4	1.0338e-01	2.3000e-01	0	$Q \in \mathcal{COP}$	3.0500e-01
Q_4	3	-2.3906e-01	2.3000e-01	7	$Q \in \mathcal{COP}$	2.4199e-01
Q_5	5	-2.6745e+00	-1.1091e-08	23	$Q \in \mathcal{COP}$	6.4999e-01
Q_6	5	-1.0000e+00	-1.9980e-08	21	$Q \in \mathcal{COP}$	4.1400e-01
Q_7	7	-3.4315e-01	-1.9960e-08	27	$Q \in \mathcal{COP}$	5.3200e-01

Table 11: Performance of the B&B algorithm for testing copositivity of Q_1 - Q_7 .

As already stated, StQP (1) can be also used to find the maximum clique of a graph $G = (V, R)$ with node set $V = \{1, \dots, n\}$ and edge set R . A clique C is defined as a subset of V such that every pair of nodes in C is connected by an edge in R . A clique C is said to be a maximum clique if it contains the largest number of nodes among all cliques, and its size $\omega(G)$ is called the (maximum) clique number. The maximum clique problem can be written as the following copositive optimization problem [34]

$$\omega(G) = \min\{\sigma \in \mathbb{N} \mid \sigma(E - D_G) - E \in \mathcal{COP}\}, \quad (63)$$

with E being the n -dimensional matrix of all-ones and $D_G = [d_{ij}]_{i,j}$ being the adjacency matrix of the graph G , i.e., $d_{ij} = 1$ if $\{i, j\} \in R$, and $d_{ij} = 0$ otherwise for all $i, j \in \{1, \dots, n\}$. So for $\sigma = \omega(G) - 1$, $Q = \sigma(E - D_G) - E \notin \mathcal{COP}$, or equivalently the StQP

$$\min_{x \in \Delta} x^T (\sigma(E - D_G) - E)x \quad (64)$$

has a negative global optimal value. Then, finding an SP of StQP (64) with a negative function value allows us to conclude that this matrix is not copositive.

We considered small and large adjacency matrices from DIMACS collection [16], and generated by [38].

In Table 12, we report the results of small instances from DIMACS collection with $\sigma = \omega(G) - 1$ obtained by applying Algorithm 5. For all the test problems, a conclusion about the copositivity of the corresponding matrices was established. For all the instances, Algorithm 5 terminates in Step 0 as the computed upper bound is negative. Then the matrices are declared to be not copositive, as expected.

The B&B algorithm faces numerical troubles in its Initialization Phase for five test problems and is not able to give a copositive certificate for the corresponding matrices. For the remaining matrices the B&B algorithm gives a certificate of copositivity by computing a global optimal solution for the corresponding StQPs. It is interesting to see that the upper bound computed in Step 0 of Algorithm 5 is the global optimal value for all these instances.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	test	CPU
c-fat14-1	14	-2.6250e-01	-1.6667e-01	-	np1	np2	-	-	-	-	-1.6667e-01	$Q \notin COP$	2.5633e-01
Brock14	14	-3.7477e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.3327e-01
Brock16	16	-3.3697e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	4.4794e-01
Brock18	18	-4.8566e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	5.5470e-01
Brock20	20	-5.4426e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	5.9638e-01
Morgen14	14	-4.2857e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	5.0205e-01
Morgen16	16	-3.7396e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	4.2047e-01
Morgen18	18	-4.7396e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	6.5630e-01
Morgen20	20	-5.4501e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.5639e-01
Morgen22	22	-4.2669e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.6509e-01
Johnson6-2-4	15	-7.3333e-01	-3.3333e-01	-	np1	np2	-	-	-	-	-3.3333e-01	$Q \notin COP$	2.5481e-01
Johnson6-4-4	15	-7.3333e-01	-3.3333e-01	-	np1	np2	-	-	-	-	-3.3333e-01	$Q \notin COP$	2.4275e-01
Johnson7-2-4	21	-8.0952e-01	-3.3333e-01	-	np1	np2	-	-	-	-	-3.3333e-01	$Q \notin COP$	2.6012e-01
Jagota14	14	-2.0154e-01	-1.6667e-01	-	np1	np2	-	-	-	-	-1.6667e-01	$Q \notin COP$	2.8346e-01
Jagota16	16	-1.2500e-01	-1.2500e-01	-	np1	np2	-	-	-	-	-1.2500e-01	$Q \notin COP$	3.0587e-01
Jagota18	18	-1.0000e-01	-1.0000e-01	-	np1	np2	-	-	-	-	-1.0000e-01	$Q \notin COP$	3.0492e-01
sanchis14	14	-3.4243e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.2105e-01
sanchis16	16	-5.0000e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.7275e-01
sanchis18	18	-5.5556e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	2.9353e-01
sanchis20	20	-6.0000e-01	-2.0000e-01	-	np1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.4803e-01
sanchis22	22	-6.3636e-01	-2.0000e-01	-	snp1	np2	-	-	-	-	-2.0000e-01	$Q \notin COP$	3.9897e-01

Table 12: Performance of Algorithm 5 for testing copositivity of Q of StQP(64) with $\sigma = \omega(G) - 1$ for small instances from DIMACS collection.

Problem	n	l_Q	opt	Nodes	test	CPU
c-fat14-1	14	-1.0000e+00	-1.6667e-01	0	$Q \notin COP$	7.6610e-01
Brock14	14	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	7.8139e-01
Brock16	16	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	7.8110e-01
Brock18	18	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	6.8200e-01
Brock20	20	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	6.9080e-01
Morgen14	14	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	6.0310e-01
Morgen16	16	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	7.4120e-01
Morgen18	18	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	8.1210e-01
Morgen20*	20	-	-	-	[gu] ?	3.0820e+00
Morgen22	22	-1.0000e+00	-3.3333e-01	0	$Q \notin COP$	8.7899e-01
Johnson6-2-4	15	-1.0000e+00	-3.3333e-01	0	$Q \notin COP$	8.3800e-01
Johnson6-4-4	15	-1.0000e+00	-3.3333e-01	0	$Q \notin COP$	8.1300e-01
Johnson7-2-4*	21	-	-	-	[gu] ?	6.8599e-01
Jagota14*	14	-	-	-	[gu] ?	9.3400e-01
Jagota16*	16	-	-	-	[gu] ?	7.3499e-01
Jagota18*	18	-	-	-	[gu] ?	7.3499e-01
sanchis14	14	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	8.9599e-01
sanchis16	16	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	2.0750e+00
sanchis18	18	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	7.6000e-01
sanchis20	20	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	7.5000e-01
sanchis22	22	-1.0000e+00	-2.0000e-01	0	$Q \notin COP$	1.9430e+00

Table 13: Performance of the B&B algorithm for testing copositivity of Q of StQP (64) with $\sigma = \omega(G) - 1$ for small instances from DIMACS collection.

The numerical results showing the performance of Algorithm 5 and the B&B algorithm with matrices of larger orders are presented in Tables 14 and 15, respectively. For this set of test problems, we set the maximum number of iterations for the enumerative algorithm in Phase 1 to be equal to 200. Moreover, we set $\alpha = 10^{-2}$ in our algorithm. For all the test problems but two, a conclusion about the copositivity of the corresponding matrices was established by Algorithm 5. For these two failures, none of the two phases of Algorithm 5 is able to improve the initial nonnegative upper bound due to the limited CPU time. Then the copositivity test is not conclusive. For six instances, the two-phase sequential algorithm terminates in Step 0 as the computed upper bound is negative. Then the matrices are declared to be not copositive. For the instance ‘Mann-a9’, the enumerative algorithm in Phase 1 is called twice. In the first call of the algorithm, a new SP with a smaller function value is computed, and in the second call, Phase 1 gives a certificate of global optimality. Hence, Phase 2 is not required. Since the optimal value is negative, then the corresponding matrix is declared to be not copositive.

The results displayed in Table 15 indicate that the B&B algorithm is unable to give a copositivity certificate for the same two matrices where Algorithm 5 failed to give one. Furthermore, the B&B algorithm failed to give such a certificate for five more matrices. The maximum computational time allowed is reached for four instances, whereas the algorithm terminates with a memory issue for the other three problems.

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	test	CPU
Brock200-1	200	-6.5666e-01	7.1798e-13	-2.4061e-05	np2	1	[1]	-	-	-	-2.4061e-05	$Q \notin COP$	1.7805e+01
Brock200-2*	200	-8.9000e-01	2.1362e-13	7.6424e-13	-	3	[1 1 200]	1	[123]	-	-	?	7.2480e+03
Brock200-3*	200	-5.5876e-01	7.6923e-02	7.6923e-02	-	1	[200]	1	[189]	-	-	?	7.4647e+03
Brock200-4	200	-5.9504e-01	6.6667e-02	-2.2355e-06	np2	1	[13]	-	-	-	-2.2355e-06	$Q \notin COP$	6.7230e+01
c-fat200-1	200	-8.7532e-01	-8.3334e-02	np1	np2	-	-	-	-	-	-8.3334e-02	$Q \notin COP$	8.9532e-01
c-fat200-2	200	-7.1371e-01	-4.1667e-02	np1	np2	-	-	-	-	-	-4.1667e-02	$Q \notin COP$	8.8655e-01
c-fat200-5	200	-3.4536e-01	-1.7241e-02	np1	np2	-	-	-	-	-	-1.7241e-02	$Q \notin COP$	8.4581e-01
hamming6-2	64	-3.1250e-02	4.0909e-01	-3.1250e-02	np2	1	[0]	-	-	-	-3.1250e-02	$Q \notin COP$	4.9620e-01
hamming6-4	64	-9.0625e-01	-2.5000e-01	np1	np2	-	-	-	-	-	-2.5000e-01	$Q \notin COP$	3.0529e-01
hamming8-2	256	-7.8125e-03	-7.8125e-03	np1	np2	-	-	-	-	-	-7.8125e-03	$Q \notin COP$	8.8724e-01
hamming8-4	256	-8.8281e-01	-6.2500e-02	np1	np2	-	-	-	-	-	-6.2500e-02	$Q \notin COP$	9.3569e-01
Johnson8-2-4	28	-4.6429e-01	1.2500e-01	-1.0000e-01	np2	2	[1 0]	-	-	-	-1.0000e-01	$Q \notin COP$	4.5640e-01
Johnson8-4-4	70	-5.8571e-01	1.1667e+00	-7.1429e-02	np2	1	[2]	-	-	-	-7.1429e-02	$Q \notin COP$	7.6972e-01
Johnson16-2-4	120	-7.0833e-01	5.0862e-01	-4.5455e-02	np2	1	[0]	-	-	-	-4.5455e-02	$Q \notin COP$	1.2295e+00
Johnson32-2-4	496	-8.4879e-01	7.3781e-01	-6.2500e-02	np2	1	[0]	-	-	-	-6.2500e-02	$Q \notin COP$	1.3419e+02
Keller4	171	-8.8304e-01	4.2857e-01	-9.0909e-02	np2	1	[25]	-	-	-	-9.0909e-02	$Q \notin COP$	4.5253e+01
Mann-a9	45	-3.3333e-01	2.9606e-17	-6.2500e-02	np2	2	[0 9]	-	-	-	-6.2500e-02	$Q \notin COP$	1.3182e+00
Mann-27	378	-3.3862e-01	5.9322e-02	-5.7902e-06	np2	1	[86]	-	-	-	-5.7902e-06	$Q \notin COP$	1.3698e+03

Table 14: Performance of Algorithm 5 for testing copositivity of Q of StQP (64) with $\sigma = \omega(G) - 1$ for large instances from DIMACS collection.

Problem	n	l_Q	opt	Nodes	test	CPU
Brock200-1	200	-1.0000e+00	-1.4638e-05	0	$Q \notin COP$	3.8555e+01
Brock200-2*	200	-1.0000e+00	9.9983e-02	112839	?	7.2002e+03
Brock200-3*	200	-1.0000e+00	7.6907e-02	64783	?	7.2006e+03
Brock200-4*	200	-1.0000e+00	6.6650e-02	31755	?	7.2004e+03
c-fat200-1	200	-1.0000e+00	-8.3370e-02	0	$Q \notin COP$	1.4490e+00
c-fat200-2	200	-1.0000e+00	-4.1739e-02	0	$Q \notin COP$	1.9400e+00
c-fat200-5	200	-1.0000e+00	-1.7367e-02	0	$Q \notin COP$	2.1440e+00
hamming6-2	64	-1.0000e+00	-3.1253e-02	0	$Q \notin COP$	7.7099e-01
hamming6-4	64	-1.0000e+00	-2.5000e-01	0	$Q \notin COP$	8.1399e-01
hamming8-2	256	-1.0000e+00	-7.8302e-03	0	$Q \notin COP$	2.2520e+00
hamming8-4	256	-1.0000e+00	-6.2523e-02	0	$Q \notin COP$	6.2850e+00
Johnson8-2-4	28	-1.0000e+00	-6.2500e-02	0	$Q \notin COP$	6.1500e-01
Johnson8-4-4	70	-1.0000e+00	-7.1431e-02	0	$Q \notin COP$	1.9050e+00
Johnson16-2-4*	120	-1.0000e+00	3.6111e-01	1996	? [gu: ofm]	1.1912e+02
Johnson32-2-4*	496	-1.0000e+00	6.4249e-01	10481	? [gu: ofm]	4.3892e+03
Keller4*	171	-1.0000e+00	0.0000e+00	120707	?	7.2002e+03
Mann-a9	45	-1.0000e+00	-6.2500e-02	0	$Q \notin COP$	1.4740e+00
Mann-27*	378	-1.0000e+00	5.9319e-02	12014	? [gu: ofm]	5.1789e+03

Table 15: Performance of the B&B algorithm for testing copositivity of Q of StQP (64) with $\sigma = \omega(G) - 1$ for large instances from DIMACS collection.

We have also generated another set of test problems that differs from the previous one on considering $\sigma = \omega(G)$ instead of $\omega(G) - 1$. Note that the resulting matrix Q of the StQP (64) is copositive and not strictly copositive for $\sigma = \omega(G)$. This implies that StQP (64) has a global optimal solution with zero optimal value.

We considered again small instances from DIMACS collection and we solved them by using Algorithm 5. The results are shown in Table 16. For all the test problems, a conclusion about the copositivity of the corresponding matrices was established. For two instances, Algorithm 5 guarantees a global minimum in Step 0 as the difference between the computed upper and lower bounds is smaller than or equal to the tolerance. Since the values of these bounds are null, then the corresponding matrices are copositive. For sixteen test problems, the enumerative algorithm in Phase 1 is called once, but it is not able to give a

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	test	CPU
c-fat14-1	14	-1.4286e-01	-3.6932e-17		-3.6932e-17	-3.6932e-17	1	[100]	1	[4]	-3.6932e-17	$Q \in COP$	4.5006e+00
Brock14	14	-2.8117e-01	9.7811e-16		9.7811e-16	9.7811e-16	1	[100]	1	[12]	9.7811e-16	$Q \in COP$	4.2493e+00
Brock16	16	-2.5704e-01	5.3474e-15		5.3474e-15	5.3474e-15	1	[100]	1	[16]	5.3474e-15	$Q \in COP$	4.8890e+00
Brock18	18	-3.2864e-01	8.1649e-15		8.1649e-15	8.1649e-15	1	[100]	1	[32]	8.1649e-15	$Q \in COP$	8.3837e+00
Brock20	20	-3.3316e-01	5.4083e-15		5.4083e-15	5.4083e-15	1	[100]	1	[46]	5.4083e-15	$Q \in COP$	1.3748e+01
Morgen14	14	-2.8551e-01	3.2025e-16		3.2025e-16	3.2025e-16	1	[100]	1	[12]	3.2025e-16	$Q \in COP$	4.8646e+00
Morgen16	16	-2.8329e-01	7.7252e-15		7.7252e-15	7.7252e-15	1	[100]	1	[14]	7.7252e-15	$Q \in COP$	5.7398e+00
Morgen18	18	-3.0229e-01	5.6185e-15		5.6185e-15	5.6185e-15	1	[100]	1	[16]	5.6185e-15	$Q \in COP$	7.1364e+00
Morgen20	20	-3.2037e-01	3.3811e-15		3.3811e-15	3.3811e-15	1	[100]	1	[16]	3.3811e-15	$Q \in COP$	7.7931e+00
Morgen22	22	-1.9216e-01	9.3555e-15		9.3555e-15	9.3555e-15	1	[100]	1	[14]	9.3555e-15	$Q \in COP$	9.7637e+00
Johnson6-2-4	15	-2.6667e-01	3.6364e-01		1.1833e-29	1.1833e-29	3	[3 0 100]	1	[34]	1.1833e-29	$Q \in COP$	5.5478e+00
Johnson6-4-4	15	-2.6667e-01	3.6364e-01		-1.8504e-17	-1.8504e-17	3	[3 0 100]	1	[34]	-1.8504e-17	$Q \in COP$	5.1785e+00
Johnson7-2-4	21	-3.8095e-01	2.3529e-01		1.9234e-15	1.9234e-15	3	[3 6 100]	1	[122]	1.9234e-15	$Q \in COP$	2.3351e+01
Jagota14	14	-7.9189e-02	1.2753e-14		1.2753e-14	1.2753e-14	1	[100]	1	[0]	1.2753e-14	$Q \in COP$	3.4846e+00
Jagota16	16	0.0000e+00	3.0997e-14	gn	np1	np2	-	-	-	-	3.0997e-14	$Q \in COP$	2.5657e-01
Jagota18	18	6.6613e-16	3.9497e-14	gn	np1	np2	-	-	-	-	3.9497e-14	$Q \in COP$	3.3670e-01
sanchis14	14	-2.5687e-01	1.3889e-15		1.3889e-15	1.3889e-15	1	[100]	1	[8]	1.3889e-15	$Q \in COP$	4.6145e+00
sanchis16	16	-2.3319e-01	1.5551e-15		1.5551e-15	1.5551e-15	1	[100]	1	[6]	1.5551e-15	$Q \in COP$	4.5696e+00
sanchis18	18	-4.4444e-01	1.9481e-15		1.9481e-15	1.9481e-15	1	[100]	1	[6]	1.9481e-15	$Q \in COP$	5.1598e+00
sanchis20	20	-5.0000e-01	5.3549e-15		5.3549e-15	5.3549e-15	1	[100]	1	[10]	5.3549e-15	$Q \in COP$	5.8213e+00
sanchis22	22	-5.4545e-01	2.3335e-15		2.3335e-15	2.3335e-15	1	[100]	1	[8]	2.3335e-15	$Q \in COP$	7.2289e+00

Table 16: Performance of Algorithm 5 for testing copositivity of Q of StQP (64) with $\sigma = \omega(G)$ for small instances from DIMACS collection.

Problem	n	l_Q	opt	Nodes	test	CPU
c-fat14-1	14	-1.0000e+00	-4.9992e-07	99	$Q \in COP$	1.6970e+00
Brock14	14	-1.0000e+00	-2.9992e-07	143	$Q \in COP$	6.5850e+00
Brock16	16	-1.0000e+00	-3.5989e-07	169	$Q \in COP$	3.3060e+00
Brock18	18	-1.0000e+00	-4.1987e-07	255	$Q \in COP$	5.4630e+00
Brock20	20	-1.0000e+00	-5.5985e-07	293	$Q \in COP$	6.9490e+00
Morgen14	14	-1.0000e+00	-2.7991e-07	171	$Q \in COP$	3.5930e+00
Morgen16	16	-1.0000e+00	-3.9989e-07	153	$Q \in COP$	3.9240e+00
Morgen18	18	-1.0000e+00	-6.1987e-07	159	$Q \in COP$	4.1960e+00
Morgen20	20	-1.0000e+00	-8.5985e-07	177	$Q \in COP$	2.6640e+00
Morgen22*	22	-	-	-	[gu] ?	1.0940e+00
Johnson6-2-4	15	-1.0000e+00	-2.3988e-07	127	$Q \in COP$	2.3370e+00
Johnson6-4-4	15	-1.0000e+00	-2.3988e-07	131	$Q \in COP$	2.4060e+00
Johnson7-2-4	21	-1.0000e+00	-2.3984e-07	379	$Q \in COP$	5.7370e+00
Jagota14*	14	-	-	-	[gu] ?	9.1700e-01
Jagota16*	16	-	-	-	[gu] ?	8.0000e-01
Jagota18*	18	-	-	-	[gu] ?	9.5000e-01
sanchis14	14	-1.0000e+00	-3.9991e-07	161	$Q \in COP$	4.0090e+00
sanchis16	16	-1.0000e+00	-5.3989e-07	157	$Q \in COP$	2.6520e+00
sanchis18	18	-1.0000e+00	-6.3987e-07	191	$Q \in COP$	3.2350e+00
sanchis20	20	-1.0000e+00	-7.1985e-07	281	$Q \in COP$	4.5440e+00
sanchis22	22	-1.0000e+00	-8.3983e-07	311	$Q \in COP$	4.5180e+00

Table 17: Performance of the B&B algorithm for testing copositivity of Q of StQP (64) with $\sigma = \omega(G)$ for small instances from DIMACS collection.

certificate of global optimality. Then Phase 2 is called once and gives this certificate. For the other three instances, the enumerative algorithm in Phase 1 is called three times and improves the initial upper bound, but it cannot give the certificate of global optimality. This certificate is given in Phase 2. The B&B algorithm was also able to compute a global minimum and give a certificate of global optimality for all these test problems but four. These failures are due to numerical troubles in the initialization phase as shown in Table 17.

Finally, the copositivity of Q in (64) with $\sigma = \omega(G)$ for large instances is tested and the results are shown in Tables 18 and 19 for Algorithms 5 and B&B Algorithm, respectively. As before, we set the maximum number of iterations for enumerative algorithm in Phase 1 equal to 200 and we use $\alpha = 10^{-2}$. As shown by the results obtained by both the algorithms, this set of problems is quite challenging. Table 18 shows that the two-phase sequential algorithm correctly verifies the copositivity of ten matrices. For two instances, Algorithm 5 guarantees a global minimum in Step 0 as the difference between the computed upper and lower bounds is smaller than or equal to the tolerance. Since the values of these bounds are null, then the corresponding matrices are copositive. The enumerative algorithm in Phase 1 is called more than once for seven test problems, but it is not able to give a certificate of global optimality. Then Phase 2 is called once

Problem	n	l_Q	u_Q	min	f_{ph1}	f_{ph2}	callEn1	ItEn1	callEn2	ItEn2	opt	test	CPU
Brock200-1*	200	-6.5105e-01	5.0000e-02	-	5.0000e-02	-	1	[200]	1	[392]	5.0000e-02	?	7.2647e+03
Brock200-2*	200	-8.3502e-01	2.0000e-01	-	2.0000e-01	-	1	[200]	1	[249]	2.0000e-01	?	7.2360e+03
Brock200-3*	200	-5.5133e-01	1.5385e-01	-	1.5385e-01	-	1	[200]	1	[323]	1.5385e-01	?	7.2144e+03
Brock200-4*	200	-5.8866e-01	1.3333e-01	-	1.3333e-01	-	1	[200]	1	[245]	1.3333e-01	?	7.2374e+03
c-fat200-1	200	-8.1250e-01	-3.2382e-17	-	-3.2382e-17	-	1	[200]	1	[138]	-3.2382e-17	$Q \in COP$	5.2773e+03
c-fat200-2	200	-6.8146e-01	-6.4753e-17	-	-6.4753e-17	-	1	[200]	1	[165]	-6.4753e-17	$Q \in COP$	6.2108e+03
c-fat200-5	200	-3.1936e-01	3.2541e-17	-	3.2541e-17	-	1	[200]	1	[261]	3.2541e-17	$Q \in COP$	6.9248e+03
hamming6-2	64	0.0000e+00	-1.7694e-16	gm	np1	np2	-	-	-	-	-1.7694e-16	$Q \in COP$	3.1460e-01
hamming6-4	64	-8.7500e-01	-5.5469e-17	gm	-5.5469e-17	-	1	[200]	1	[323]	-5.5469e-17	$Q \in COP$	6.2026e+03
hamming8-2	256	4.4409e-16	1.6480e-16	gm	np1	np2	-	-	-	-	1.6480e-16	$Q \in COP$	7.1521e-01
hamming8-4	256	-8.7500e-01	-8.3262e-17	-	-8.3262e-17	-	1	[200]	1	[19]	-8.3262e-17	$Q \in COP$	2.2087e+03
Johnson8-2-4	28	-4.2857e-01	5.0000e-01	-	8.3267e-17	-	7	[5 1 0 6 0 1 200]	1	[282]	8.3267e-17	$Q \in COP$	8.3198e+01
Johnson8-4-4	70	-5.7143e-01	1.3333e+00	-	1.2822e-20	-	3	[1 0 200]	1	[216]	1.2822e-20	$Q \in COP$	6.2430e+03
Johnson16-2-4	120	-7.0000e-01	7.2414e-01	-	-4.1633e-17	-	3	[0 0 200]	1	[2813]	-4.1633e-17	?	7.2032e+03
Johnson32-2-4	496	-8.4677e-01	8.5366e-01	-	-1.0396e-17	-	12	[0 0 0 0 0 1 0 4 1 0 200]	1	[0]	-1.0396e-17	[gu]?	1.1662e+04
Keller4	171	-8.7135e-01	5.7143e-01	-	2.9630e-12	-	4	[19 11 10 200]	1	[683]	2.9630e-12	?	7.2078e+03
Mann-a9	45	-2.8889e-01	6.6667e-02	-	3.9669e-14	-	2	[0 200]	1	[498]	3.9669e-14	$Q \in COP$	3.5659e+03
Mann-27	378	-3.3333e-01	6.7797e-02	-	7.9947e-03	-	4	[6 0 40 200]	1	[0]	7.9947e-03	[gu]?	3.0886e+03

Table 18: Performance of Algorithm 5 for testing copositivity of Q of StQP (64) with $\sigma = \omega(G)$ for large instances from DIMACS collection.

Problem	n	l_Q	opt	Nodes	test	CPU
Brock200-1*	200	-1.0000e+00	4.9984e-02	20669	?	7.2004e+03
Brock200-2*	200	-1.0000e+00	1.9998e-01	106191	?	7.2004e+03
Brock200-3*	200	-1.0000e+00	1.5383e-01	60413	?	7.2003e+03
Brock200-4*	200	-1.0000e+00	1.3332e-01	41237	?	7.2009e+03
c-fat200-1	200	-1.0000e+00	-4.0038e-05	1731	$Q \notin COP$	8.0240e+01
c-fat200-2	200	-1.0000e+00	-7.5678e-05	1593	$Q \notin COP$	1.1437e+02
c-fat200-5	200	-1.0000e+00	-1.2882e-04	1847	$Q \notin COP$	2.9362e+02
hamming6-2*	64	-1.0000e+00	-3.1997e-06	442239	?	7.2000e+03
hamming6-4	64	-1.0000e+00	0.0000e+00	3017	$Q \in COP$	8.7308e+01
hamming8-2*	256	-1.0000e+00	-1.7918e-05	4191	?	7.2032e+03
hamming8-4*	256	-1.0000e+00	-2.4638e-05	12023	?	7.2027e+03
Johnson8-2-4	28	-1.0000e+00	0.0000e+00	1191	$Q \in COP$	2.3916e+01
Johnson8-4-4*	70	-1.0000e+00	-3.3594e-06	394529	?	7.2000e+03
Johnson16-2-4*	120	-1.0000e+00	0.0000e+00	77125	?	7.2003e+03
Johnson32-2-4*	496	-1.0000e+00	0.0000e+00	6805	?	7.2059e+03
Keller4*	171	-1.0000e+00	0.0000e+00	25677	?	7.2006e+03
Mann-a9*	45	-1.0000e+00	-4.5979e-07	232189	?	7.2002e+03
Mann-27*	378	-1.0000e+00	6.7794e-02	13841	?	7.2012e+03

Table 19: Performance of the B&B algorithm for testing copositivity of Q of StQP (64) with $\sigma = \omega(G)$ for large instances from DIMACS collection.

and gives this certificate for three of them. For the first four test problems the two phases of Algorithm 5 cannot improve the positive upper bound computed in the initialization step and cannot give a certificate of global optimality. Numerical troubles are encountered in GUROBI when the instances ‘Johnson16-2-4’ and Mann-27 are solved.

The B&B algorithm encounters many difficulties in solving this set of problems. For six instances, the algorithm computes a global minimum with a null global optimal value, but for only two of them, it gives the desired certificate of global optimality that provides the correct conclusion about the copositivity. For three instances an approximate solution is computed by IPOPT [37] in the initialization phase and it is not improved over the iterations. So, the algorithm is not able to compute a correct global minimum and this leads to an incorrect conclusion about the copositivity of the corresponding matrices. For all other instances, the algorithm could not terminate in the allowed computational time and the copositivity test is inconclusive.

As a final conclusion of the experiments for investigating the copositivity of some matrices, the two-phase sequential algorithm is shown to be efficient for solving almost all the required StQPs. Phase 1 of the algorithm failed to compute an SP with a negative objective value for the StQPs associated with two matrices that are not copositive. Furthermore, the algorithm also failed to compute a global minimum for eight StQPs associated to matrices that are copositive. The results of the experiments of applying the B&B algorithm to the same matrices show that this algorithm has a bigger number of failures for giving a copositivity certificate than the two-phase algorithm.

To end this section, we illustrate the overall comparison of the computational effort in terms of CPU time of Algorithm 5 and the B&B algorithm, we show in Fig. 1 their performance profiles for all instances considered in this paper.

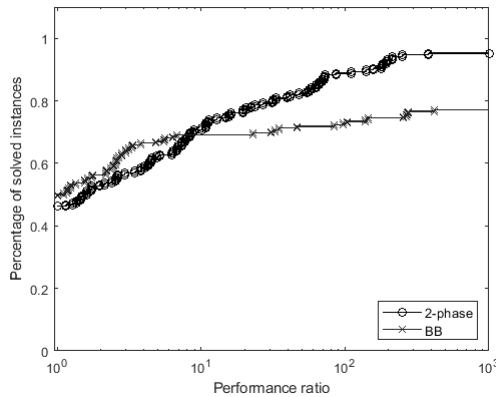


Fig. 1: Performance profile of Algorithm 5 and B&B algorithm for all the instances considered in this paper.

The percentage of instances solved by the two-phase sequential and B&B algorithms amounts to 95% and 77%, respectively. As far as the CPU is concerned, for more than half of the instances the B&B algorithm is faster than the proposed algorithm. Note that this difference of time essentially depends on the big time that the local NLP solver takes for solving the NLPs required by Phase 2.

9 Conclusions

In this paper, we discuss a new two-phase methodology for global optimization of the Standard Quadratic Optimization Problem (StQP). The first phase is devoted to the computation of Stationary Points (SP) with strictly decreasing function values, while the second phase is concerned with a certificate that establishes that the last SP computed in Phase 1 is a global minimum. The algorithm is shown to be efficient for its goal and seems to be competitive with the best state-of-art Branch and Bound algorithm for the StQP.

The efficiency of the two-phase algorithm depends very much on the nonlinear programming (NLP) techniques that are employed in the enumerative algorithms used in the two phases. We believe that a more efficient algorithm may be designed for solving a convex NLP with a linear function and linear and convex quadratic constraints, that is required in Phase 2. This topic is an important topic of our future research.

The novel two-phase idea for the StQP introduced in this paper can be implemented for a general Linear Program with Linear Complementarity Constraints (LPLCC) and for a number of important global optimization problems with linear constraints that are equivalent to LPLCC, such as the popular Quadratic Programming Problem and the Linear Bilevel Programming Problem. These are also important topics of our future research.

Acknowledgements The research of Valentina Sessa benefited from the support of the FMJH Program PGMO and from the support of EDF.

The research of Joaquim Júdice was partially supported in the scope of R&D Unit UID/EEA/50008/2019, financed by the applicable financial framework (FCT/MEC) through national funds and when applicable co-funded by FEDER-PT2020 Partnership Agreement.

References

1. Best, M.J., Ding, B.: Global and local quadratic minimization. *J Glob Optim* **10**(1), 77–90 (1997). DOI 10.1023/A:1008278114178
2. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. *SIAM Rev.* **59**(1), 65–98 (2017). DOI 10.1137/141000671
3. Bomze, I.M.: On standard quadratic optimization problems. *J Glob Optim* **13**(4), 369–387 (1998). DOI 10.1023/A:1008369322970
4. Bomze, I.M.: Branch-and-bound approaches to standard quadratic optimization problems. *J Glob Optim* **22**(1/4), 17–37 (2002). DOI 10.1023/A:1013886408463
5. Bomze, I.M., De Klerk, E.: Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *J Glob Optim* **24**(2), 163–185 (2002). DOI 10.1023/A:1020209017701
6. Bomze, I.M., Dur, M., De Klerk, E., Roos, C., Quist, A.J., Terlaky, T.: On copositive programming and standard quadratic optimization problems. *J Glob Optim* **18**(4), 301–320 (2000). DOI 10.1023/A:1026583532263
7. Bomze, I.M., Eichfelder, G.: Copositivity detection by difference-of-convex decomposition and ω -subdivision. *Math. Program.* **138**(1-2), 365–400 (2013). DOI 10.1007/s10107-012-0543-x
8. Bomze, I.M., Frommlet, F., Rubey, M.: Improved SDP bounds for minimizing quadratic functions over the ℓ_1 -ball. *Optimization Letters* **1**(1), 49–59 (2006). DOI 10.1007/s11590-006-0018-1
9. Bomze, I.M., Locatelli, M., Tardella, F.: New and old bounds for standard quadratic optimization: Dominance, equivalence and incomparability. *Math. Program.* **115**(1), 31–64 (2008). DOI 10.1007/s10107-007-0138-0
10. Bomze, I.M., Palagi, L.: Quartic formulation of standard quadratic optimization problems. *J Glob Optim* **32**(2), 181–205 (2005). DOI 10.1007/s10898-004-2701-z
11. Bomze, I.M., Schachinger, W., Ullrich, R.: The complexity of simple models—A study of worst and typical hard cases for the standard quadratic optimization problem. *Math. Oper. Res.* **43**(2), 651–674 (2018). DOI 10.1287/moor.2017.0877
12. Bonami, P., Lodi, A., Schweiger, J., Tramontani, A.: Solving quadratic programming by cutting planes. *SIAM J. Optim.* **29**(2), 1076–1105 (2019). DOI 10.1137/16M107428X
13. Brás, C., Eichfelder, G., Júdice, J.: Copositivity tests based on the linear complementarity problem. *Comput Optim Appl* **63**(2), 461–493 (2016). DOI 10.1007/s10589-015-9772-2
14. Cottle, R., Pang, J., Stone, R.: *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics (2009)
15. Currie, J., Wilson, D.I.: OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user. In: *Foundations of Computer-Aided Process Operations*. Georgia, USA (2012)
16. DIMACS: Second DIMACS Challenge. Test instances available at <http://dimacs.rutgers.edu/challenges>. <http://dimacs.rutgers.edu/challenges>
17. Fernandes, L., Júdice, J., Sherali, H., Fukushima, M.: On the computation of all eigenvalues for the eigenvalue complementarity problems. *J Glob Optim* **59**, 307–326 (2014)
18. Fletcher, R., Leyfer, S.: Nonlinear programming without a penalty function. *Math. Program.* **91**(2), 239–269 (2002). DOI 10.1007/s101070100244
19. Furini, F., Traversi, E., Belotti, P., Frangioni, A., Gleixner, A., Gould, N., Liberti, L., Lodi, A., Misener, R., Mittelmann, H., Sahinidis, N.V., Vigerske, S., Wiegele, A.: QPLIB: A library of quadratic programming instances. *Math. Prog. Comp.* **11**(2), 237–265 (2019). DOI 10.1007/s12532-018-0147-4
20. Gondzio, J., Yıldırım, E.A.: Global solutions of nonconvex standard quadratic programs via mixed integer linear programming reformulations. *J Glob Optim* (2021). DOI 10.1007/s10898-021-01017-y
21. Gurobi Optimization, L.: Gurobi optimizer reference manual. <http://www.gurobi.com>
22. Hall, M., Newman, M.: Copositive and completely positive quadratic forms. *Math. Proc. Camb. Phil. Soc.* **59**(2), 329–339 (1963). DOI 10.1017/S0305004100036951
23. Hoffman, A.J., Pereira, F.: On copositive matrices with -1, 0, 1 entries. *J. Comb. Theory Ser. A.* **14**(3), 302–309 (1973). DOI 10.1016/0097-3165(73)90006-X
24. Horst, R., Pardalos, P.M., Thoai, N.V.: *Introduction to Global Optimization*, 2nd ed edn. No. v. 48 in *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht ; Boston (2000)
25. Júdice, J.J., Fukushima, M., Iusem, A., Martinez, J.M., Sessa, V.: An alternating direction method of multipliers for the eigenvalue complementarity problem. *Optim. Methods Softw.* **36**(2-3), 337–370 (2021). DOI 10.1080/10556788.2020.1734804
26. Kaplan, W.: A copositivity probe. *Linear Algebra Appl.* **337**(1-3), 237–251 (2001). DOI 10.1016/S0024-3795(01)00351-2
27. Liuzzi, G., Locatelli, M., Piccialli, V.: A new branch-and-bound algorithm for standard quadratic programming problems. *Optim. Methods Softw.* **34**(1), 79–97 (2019). DOI 10.1080/10556788.2017.1341504
28. MatrixMarket: math.nist.gov/MatrixMarket/index.html. math.nist.gov/MatrixMarket/index.html
29. Meurant, G.A.: Computer Solution of Large Linear Systems, 1. ed edn. No. 28 in *Studies in Mathematics and Its Applications*. Elsevier, Amsterdam (1999)
30. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. *Can. J. Math.* **17**, 533–540 (1965). DOI 10.4153/CJM-1965-053-6
31. Nowak, I.: A Global Optimality Criterion for Nonconvex Quadratic Programming over a Simplex. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik (2005). DOI 10.18452/2684
32. Scozzari, A., Tardella, F.: A clique algorithm for standard quadratic programming. *Discret. Appl. Math.* **156**(13), 2439–2448 (2008). DOI 10.1016/j.dam.2007.09.020
33. Sherali, H.D., Tuncbilek, C.H.: A global optimization algorithm for polynomial programming problems using a Reformulation-Linearization Technique. *J Glob Optim* **2**(1), 101–112 (1992). DOI 10.1007/BF00121304

-
34. Sponsel, J., Bundfuss, S., Dür, M.: An improved algorithm to test copositivity. *J Glob Optim* **52**(3), 537–551 (2012). DOI 10.1007/s10898-011-9766-2
 35. Stix, V.: Target-oriented branch and bound method for global optimization. *J Glob Optim* **26**(3), 261–277 (2003). DOI 10.1023/A:1023245011830
 36. Väliäho, H.: Quadratic-programming criteria for copositive matrices. *Linear Algebra Appl.* **119**, 163–182 (1989). DOI 10.1016/0024-3795(89)90076-1
 37. Wächter, A., Biegler, L.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Program.* **106**, 25–57 (2006)
 38. Žilinskas, J., Dür, M.: Depth-first simplicial partition for copositivity detection, with an application to MaxClique. *Optim. Methods Softw.* **26**(3), 499–510 (2011). DOI 10.1080/10556788.2010.544310