J Glob Optim (2009) 45:281–296
DOI 10.1007/s10898-008-9374-y

On the computation of C^* certificates

Florian Jarre · Katrin Schmallowsky

Received: 4 May 2008 / Accepted: 24 October 2008 / Published online: 12 November 2008
© Springer Science+Business Media, LLC. 2008

Abstract The cone of completely positive matrices C^* is the convex hull of all symmetric rank-1-matrices xx^T with nonnegative entries. While there exist simple certificates proving that a given matrix $B \in C^*$ is completely positive it is a rather difficult problem to find such a certificate. We examine a simple algorithm which—for a given input B—either determines a certificate proving that $B \in C^*$ or converges to a matrix $\bar{S}$ in C^* which in some sense is "close" to B. Numerical experiments on matrices B of dimension up to 200 conclude the presentation.

Keywords Completely positive matrices

1 Introduction

The concept of completely positive matrices has been introduced more than 40 years ago, see [8,10]. The interest in the cone C^* of completely positive matrices and its dual cone C of copositive matrices has recently gained new momentum in the context of combinatorial and global optimization problems. In [3], Bomze et al. use linear programs over the completely positive cone to approximate solutions of "standard quadratic optimization problems"—a class of problems that is $\mathcal{NP}$-hard. More recently, de Klerk and Pasechnik [6] pointed out that standard quadratic optimization problems can be reformulated as linear optimization problems over the cone of copositive matrices—presenting a simple equivalent reformulation, not just a relaxation. This result was generalized by Burer [5] to binary and continuous nonconvex quadratic programs.

A recent summary of theoretical properties of the cones C^* and C can be found in [2], see also [7]. Checking copositivity of a given matrix is co-$\mathcal{NP}$-complete, (see e.g. [11]), and likewise the problem of determining complete positivity of a matrix B is difficult. Berman

F. Jarre (✉) · K. Schmallowsky
Mathematisches Institut, University of Düsseldorf, Universitätsstr. 1, 40225 Düsseldorf, Germany
e-mail: jarre@opt.uni-duesseldorf.de

and Rothblum [1] establish a finite but highly exponential algorithm for this problem[1], but there is also no simple heuristic to date to approach the problem of determining complete positivity of a matrix B of moderate dimension. (In particular, the Cholesky factor and the symmetric square root typically have negative entries, and thus do not render the desired nonnegative factorization.)

This work aims at providing a practical tool for generating a certificate of complete positivity of a given matrix B—or to compute such a certificate for a "nearby" completely positive matrix. To this end the quadratic factorization heuristic of [4] is modified. The main contributions of the present paper are the introduction of a regularization step, the discussion of its implementation, and finally, some encouraging numerical examples. The numerical examples are based on the recent "augmented primal-dual" (apd) method in [9], exploiting the structure of the large scale sub-problems by a first-order method for conic optimization that quickly generates approximate solutions but that is less suitable for high accuracy solutions.

1.1 The cp-rank

By $I\!R_+$ we denote the set of nonnegative numbers, $I\!R_+ = \{t \in I\!R \mid t \geq 0\}$. The inequality $X \geq 0$ is used to indicate that the matrix X only has nonnegative entries; such X is called nonnegative. By $B \succeq 0$ we indicate that the symmetric matrix B is positive semidefinite, and write $S^n_+ := \{S = S^T \in I\!R^{n\times n} \mid S \succeq 0\}$. By C^* we denote the cone of completely positive matrices, i.e. the convex hull of all xx^T with $x \in I\!R^n_+$. (In this paper the dimension of the matrices in C^* will always be denoted by n.) By definition, for any $B \in C^*$ there exists a natural number p and a $n \times p$-matrix $X \geq 0$ such that $B = XX^T$.

For a given matrix $B \succeq 0$ the algorithm of the present paper aims at generating a matrix $X \geq 0$ such that $B = XX^T$ holds true. If the algorithm succeeds then the matrix X provides a certificate for the statement $B \in C^*$. The dimension p of the $n \times p$ matrix X is discussed next. Evidently, when $B \in C^* \subset S^n_+$, the Cholesky factor $L \in I\!R^{n\times p}$ of $B = LL^T$ can be computed with $p \leq n$. ($p < n$ when B has zero eigenvalues.) On the other hand, even when $B \in C^*$, the matrix L typically is not nonnegative, and, as discussed next, the choice $p \leq n$ is not suitable in general.

Given a matrix $B \in C^*$ the minimal number p for which there is an $n \times p$-matrix $X \geq 0$ such that $B = XX^T$ is called the cp-rank of B, see e.g. [2].

Let E be the $l \times l$ all-ones matrix and I the $l \times l$ identity matrix. Then, it is straightforward to verify that

$$\hat{S} := \begin{pmatrix} lI & E \\ E & lI \end{pmatrix}$$

has cp-rank l^2. Thus, for even numbers n there exist $n \times n$-matrices $B \in C^*$ with cp-rank $n^2/4$. In fact, also matrices nearby $\hat{S}$ have a large cp-rank: Let $U_\epsilon(\hat{S}) := \{S = S^T \mid \|S - \hat{S}\|_F \leq \epsilon\}$. Then, for sufficiently small $\epsilon > 0$, all matrices in $C^* \cap U_\epsilon(\hat{S})$ have a cp-rank of at least l^2. As is intuitively clear and confirmed by preliminary numerical experiments in Sect. 4.3, it is difficult to generate a C^*-certificate for such matrices with high cp-rank.

When n is not even, there also exist matrices with cp-rank $\geq \lfloor n^2/4 \rfloor$. On the other hand, by Caratheodory's theorem, the cp-rank of a matrix $B \in C^*$ always satisfies $p \leq n(n+1)/2$.

[1] To our knowledge, $\mathcal{NP}$-completeness of this problem has not yet been firmly established. The authors believe that it is possible to generate an approximate separation oracle for C^* given an oracle that determines whether $B \in C^*$ or not. Based on such a separation oracle, the solutions of linear optimization problems over C^* can be approximated in polynomial time, and such approximations are $\mathcal{NP}$-complete. A careful analysis of this approach is beyond the scope of this paper.

Remark 1.1 For $1 \leq p < \lfloor n^2/4 \rfloor$ the set $C_p^* := \{XX^T \mid X \in I\!R^{n\times p},\ X \geq 0\}$ of matrices with cp-rank $\leq p$ is not convex.

Proof The proof is a trivial consequence of the observation that there exist matrices with cp-rank $> p$. Let S be one such matrix, then S is a convex combination of positive rank-1-matrices each of which is contained in C_p^*. □

This simple observation has implications on the selection of p in Algorithm 2.1 below.

1.2 Further notation

We use the following common notation: The scalar product of two matrices $X, \hat{X} \in I\!R^{n\times p}$ is denoted by $\langle X, \hat{X}\rangle = (X^T\hat{X}) = \sum_{i=1}^n \sum_{j=1}^p X_{i,j}\hat{X}_{i,j}$; it induces the Frobenius norm, $\|X\|_F^2 = \langle X, X\rangle$. By $x := \text{vec}\,(X) \in I\!R^{np}$ we denote the vector obtained by stacking the columns of X on top of each other, and by $X := \text{mat}(x) \in I\!R^{n\times p}$ we denote the inverse operation. The dimension of $x = \text{vec}\,(X)$ is uniquely determined by the dimension of X but the converse is not true; in this paper the dimension of mat(x) will always be clear from the context. Likewise, tuples of matrices such as (X, S) are represented by a single vector $z = \text{vec}\,((X, S))$ with the inverse operator $(X, S) = \text{mat(z)}$. By E we always denote the matrix of all ones, I the identity matrix, and e the vector of all ones. The dimensions will always be evident from the context.

1.3 Outline

Section 2 presents a linearization technique to approach a given matrix B from within C^*. Motivated by the properties of this algorithm a normalization of a matrix $B \succeq 0$ and the computation of a "central" starting point in C^* is discussed in Sect. 3. Even with a suitable starting point the approach of Sect. 2 may stagnate. Section 4 presents a heuristic (called "regularization") to recover from stagnation. Preliminary numerical experiments for the approach with and without regularization are listed in Sects. 2.6 and 4.3. As we are not aware of other approaches for solving this problem for matrices of moderate dimensions we cannot present comparisons with existing approaches.

2 A Lyapunov type second-order cone algorithm

Given a symmetric matrix $B \in I\!R^{n\times n}$ we wish to compute a matrix $S \in C^*$ that is in some sense "close" to B. In Sect. 3 we will outline how we can assume without loss of generality that the diagonal of B is all ones and that an initial approximation $X = X^0 \geq 0, X \in I\!R^{n\times p}$ is given such that $XX^T \approx B$ and XX^T is in the interior of C^*. (The approximation $XX^T \approx B$ may be poor.)

2.1 Motivation

The quadratic factorization heuristic of [4] can be adapted to the problem of generating a certificate of complete positivity: If B is in C^*, then there exists a matrix ΔX^* such that

$$(X + \Delta X^*)(X + \Delta X^*)^T = B \tag{1}$$

and $X + \Delta X^* \geq 0$. Neglecting the second-order term $\Delta X^*(\Delta X^*)^T$ in (1) we obtain the linearized equation yielding an approximation ΔX for ΔX^*:

$$X \Delta X^T + \Delta X X^T = B - X X^T. \tag{2}$$

For a given $B \in C^*$ the set of ΔX^* satisfying (1) contains more than one element. The fact that the linearization error in (2) depends on $\|\Delta X\|_F$ suggests to determine an approximation ΔX for ΔX^* based on the linearized problem

$$\text{minimize } \|\Delta X\|_F \mid X \Delta X^T + \Delta X X^T = R, \quad X + \Delta X \geq 0, \tag{3}$$

where $R = B - X X^T$. When $X > 0$ has full rank the Lyapunov equation (2) has a solution ΔX, and problem (3) has a solution for small $\|R\|$.[2] Problem (3) is the basis for an iterative process with repeated updates of the form $X \mapsto X + \alpha \Delta X$ with $\alpha \in (0, 1]$.

Problem (3) is a second-order cone program (SOC problem) with np variables and $n(n+1)/2$ equality constraints. For small size problems, the subproblems (3) can be solved by interior-point approaches. The linear operator defining the equality constraints is not given by a matrix and its explicit representation by a matrix (mapping mat(ΔX) to mat(R) induces large storage requirements. Thus, interior point methods such as SeDuMi [14] prove to be less suitable for this type of problem when the dimension of ΔX is moderate or large, say $n \cdot p \geq 1000$, see Sect. 2.6. To be able to handle problems of the form (3) with a large number of variables and constraints, a specialized approach is discussed next that does not require a specific representation of the equality constraints.

2.2 Reformulation of the second-order cone program

The augmented primal dual (apd) method of [9,13] is applied to conic programs in a certain standard form due to Nesterov and Nemirovskii [12],

$$(P) \quad \text{minimize } \langle c, x \rangle \text{ s.t. } x \in K \cap (\mathcal{L} + b),$$

and

$$(D) \quad \text{minimize } \langle b, s \rangle \text{ s.t. } s \in K^D \cap (\mathcal{L}^\perp + c),$$

where K is a closed convex cone in a finite dimensional Euclidean space, and $\mathcal{L}$ is a linear subspace. The apd method uses projections onto the affine subset $(\mathcal{L} + b) \times (\mathcal{L}^\perp + c)$ and projections onto the primal-dual cone. It is accelerated by the use of a regularization term that exploits the complementarity condition and a limited memory BFGS approach.

We illustrate how problem (3) can be reformulated to the form (P). (3) is equivalent to the problem

$$\text{minimize } x_0 \mid x_0 \geq \|x_1\|_2, \ x_2 \geq 0, \ \mathcal{A}(X_1) = R, \ -X_1 + X_2 = X, \tag{4}$$

where $X_1 := \Delta X, x_1 := \text{vec}(X_1)$, $X_2 := X + \Delta X$, $x_2 := \text{vec}(X_2)$, and $\mathcal{A}(\Delta X) := X \Delta X^T + \Delta X X^T$ depends on the current iterate X. Recall that X is an $n \times p$-matrix where

[2] Indeed, when X has full rank it has an $n \times n$ square submatrix of full rank. Setting the entries of ΔX to zero that are not multiplied with this full rank submatrix, we assume, without loss of generality, that X and ΔX are square matrices. If $X = U \Sigma V^T$ is the singular value decomposition of X then (2) can be rewritten equivalently as $\Sigma V^T \Delta X^T U + U^T \Delta X V \Sigma = U^T R U$. Denote the new unknown by $\widetilde{\Delta X} := U^T \Delta X V$. Then, for symmetric R, this equation can easily be solved for $\widetilde{\Delta X}$ (since Σ is positive definite and diagonal). With $\Delta X := U \widetilde{\Delta X} V^T$ this proves solvability of (2). As the solution ΔX obtained in this way is continuous (linear) in R, it satisfies $X + \Delta X > 0$ when $\|R\|$ is sufficiently small.

p is some upper bound for the cp-rank of B. With the above notations we may write problem (4) in the form (P) as

$$\text{minimize } x_0 \mid x = (x_0, x_1, x_2)^T \in \mathcal{K} \cap (\mathcal{L} + \bar{R}). \tag{5}$$

Here, $\mathcal{K}$ is the cone $\mathcal{K} = \mathcal{Q}^{np+1} \times \mathbb{R}_+^{np}$ with

$$\mathcal{Q}^{np+1} := \{x := (x_0; x_1) \in \mathbb{R} \times \mathbb{R}^{np} \mid x_0 \geq \|x_1\|_2\}$$

being the second-order cone of dimension $np + 1$. The linear set $\mathcal{L}$ in (5) is given by $\mathcal{L} := \{x \ \mid \hat{A}x = 0\}$, where

$$\hat{A}x = \begin{bmatrix} 0 & A & 0 \\ 0 & -I & I \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}.$$

Here, A represents the linear operator $\mathcal{A}$ such that $Ax_1 = \text{vec}(\mathcal{A}(X_1))$ for $x_1 = \text{vec}(X_1)$. The linear equations of problem (5) can be written as

$$\hat{A}x = \begin{bmatrix} \text{vec}(R) \\ \text{vec}(X) \end{bmatrix} =: \hat{r} =: \hat{A}\bar{r}$$

for some suitable vector $\bar{r}$. This defines the element $\bar{R} = mat(\bar{r}) \in \mathbb{R} \times \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p}$ in (5).

2.3 Solution of the SOC problem

As the correction $X_1 = \Delta X$ is subject to a linearization error (resulting from (2)), the subproblems (5) are not solved up to full precision in the implementation in Sect. 2.6. Instead, these subproblems are solved iteratively, and when the accuracy obtained for the subproblem is of the same magnitude as the linearization error, the algorithm for solving the subproblem is stopped.

Since the projection of a given iterate onto the set $\mathcal{K}$ is trivial, the main computational effort in the apd-approach for solving (5) is the repeated computation of the projection of the current iterate onto the linear set $\mathcal{L}$. As detailed below, this projection is computationally cheap as well. Moreover, as the required accuracy of the approximate solution of (5) is low, the apd-method in [9,13] seems to be very well suited for solving the subproblems (5).

The projection of a point x onto $\mathcal{L}$ is given by

$$\Pi_{\mathcal{L}}(x) = x - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A}x.$$

Multiplications by $\hat{A}$ and $\hat{A}^T$ are cheap. The only critical part in the computation of this projection is the solution of a linear equation of the form

$$\hat{A}\hat{A}^T g = h \tag{6}$$

for a given right hand side $h = (h_1, h_2)^T \in \mathbb{R}^{n^2+np}$. Equation 6 is given by

$$\begin{bmatrix} AA^T & -A \\ -A^T & 2I \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{7}$$

and its solution is obtained from

$$AA^T g_1 = 2h_1 + Ah_2 =: \hat{h}.$$

(Trivially, $g_2 = \frac{1}{2}(h_2 + A^T g_1)$.) Writing this equation in operator notation (with $n \times n$ matrices $G_1 := mat(g_1)$ and $\hat{H} := mat(\hat{h})$) leads to

$$\mathcal{A}\mathcal{A}^*(G_1) = \hat{H},$$

which is precisely the following Lyapunov equation

$$XX^T G_1^T + G_1 XX^T = \hat{H}. \tag{8}$$

Below, we discuss the solution of the above Lyapunov equation for the case that X has full row rank: To this end let

$$XX^T = U\Sigma U^T$$

be the eigenvalue decomposition of XX^T, i.e. $U \in I\!R^{n\times n}$ is an orthogonal matrix, and Σ is an $n \times n$ positive definite diagonal matrix.
Denoting $\widetilde{G_1} := U^T G_1 U$ and $\tilde{H} := U^T \hat{H} U$, Eq. 8 is equivalent to

$$\Sigma \widetilde{G_1}^T + \widetilde{G_1}\Sigma = \tilde{H}. \tag{9}$$

The solution of this system is given by $\widetilde{G_1}_{i,j} = \tilde{H}/(\Sigma_{ii} + \Sigma_{jj})$ for $1 \le i, j \le n$. This yields the solution $G_1 = U\widetilde{G_1}U^T$ of (8).

2.4 Overall algorithm

Now, we summarize an algorithm based on (5):

Algorithm 2.1 *[Lyapunov type SOC algorithm]*

1. Input: A matrix $X^0 \ge 0$ of full row rank and a matrix $B = B^T$.
   Set $k := 0$, $S^0 := P^0 := X^0(X^0)^T$.
2. Set $\hat{R}^k := \frac{1}{2}(B + P^k) - S^k$.
3. Solve problem (5) for $X = X^k$ and determine a step size $\alpha_k \in (0, 1]$ such that $\|(X^k + \alpha_k \Delta X)(X^k + \alpha_k \Delta X)^T - B\| < \|S^k - B\|$.
4. Set $X^{k+1} := X^k + \alpha_k \Delta X^k$, $S^{k+1} := X^{k+1}(X^{k+1})^T$, and compute the projection P^{k+1} of S^{k+1} onto the straight line connecting S^0 and B.
5. Set $k = k + 1$ and go to Step 2.

Remarks

- If problem (5) (i.e. problem (3)) has a feasible solution ΔX then,

  $$\begin{aligned}&\|(X^k + \alpha\Delta X)(X^k + \alpha\Delta X)^T - B\| \\ &\quad = \|(1-\alpha)(S^k - B) + \alpha^2 \Delta X \Delta X^T\| < \|S^k - B\|\end{aligned}$$

  for small $\alpha > 0$. By using an exact line search in Step 3, the distance of S^{k+1} to B is strictly less than the distance of S^k to B.
- As pointed out, for small $\|R\|$, problem (3) has a solution when $X > 0$ has full rank. In this case, XX^T is in the interior of C^*. When XX^T is close to the boundary of C^*, the Eq. 2 may not have a solution of small norm $\|\Delta X\|$. Algorithm 2.1 therefore aims at keeping the iterates S^k "central" in some form. In Sect. 3 the computation of a starting

point S^0 in the "center"[3] of C^* will be addressed. When S^0 is a "center" of C^* and $B \in C^*$, then the segment $[S^0, B)$ can be viewed as a "central path" in the interior of C^* leading to B. Due to the linearization error, the iterates deviate from this "central path". The construction of P^{k+1} on the line segment $[S^0, B)$ may be regarded as a centering component. By choosing the new target as a convex combination of B and P^k in Step 2, the deviation of the iterates S^k from the line segment $[S^0, B)$ can be controlled.

- While the condition "$X > 0$ has full rank and $\|R\|$ is small" is sufficient to guarantee the existence of a solution of (3), this condition is not guaranteed in the course of the algorithm. (Computing an upper bound for $\|R\|$ that guarantees the existence of a solution would be very expensive, for example.) Our attempts to establish global convergence for sufficiently short steps failed; the algorithm is therefore intended merely as a heuristic and does not enforce the condition "$X > 0$ of full rank" either.
- As the problem under consideration is rather difficult, it is unlikely that the simple Algorithm 2.1 will generate a solution for all problem instances. In Sect. 4 we therefore present a computationally expensive regularization step to improve the convergence behavior for difficult instances.

2.5 Matrix completion

We point out that Algorithm 2.1 can be used with minor modifications to (approximately) solve the completely positive completion problem: "Given an index set $\mathcal{I} \subset \{1, \ldots, n\}^2$ and a symmetric matrix $B \in I\!R^{n\times n}$, find a matrix $S \in C^*$ such that $S_{i,j} = B_{i,j}$ for all $(i, j) \in \mathcal{I}$." In this case, the equality constraints in problems (3) or (4) that correspond to index pairs not in $\mathcal{I}$ are simply dropped. Unfortunately, the constraints then do not lend themselves any longer to the application of the apd-algorithm; the inverse of AA^T is not given by (8). For small size problems, of course, interior-point algorithms could be used in place of the apd-approach.

2.6 Random examples

Algorithm 2.1 was tested for random examples with $p = 2n$ and $n = 10, 50, 200$. The unknown matrix X is of dimension $n \times p$, and an initial matrix $X = X^0 \geq 0$ with $XX^T \approx B$ is generated as outlined in Sect. 3 below. The matrix B was generated as $B = WW^T$ where W was chosen as a random (uniformly distributed entries in (0, 1)) matrix of dimension $n \times k$. For $k < n$ it follows that $B \in \partial C^*$. We observe significantly longer solution times for the case $k < n$ with less accuracy in the final solution. The solution times reflect that the subproblems to be solved with the apd method tend to require a higher number of iterations; the number of overall (outer) iterations does not vary to such an extent.

The results of Algorithm 2.1 are given in Tables 1, 2 and 3. The running times refer to a 1.6 GZ PC (from 2003).

We also tested some examples with $B \notin C^*$. Here the convergence behavior was similar as in the case $B \in \partial C^*$; convergence slowed down at some point, but the residual was still large. Let us denote the final matrix generated by Algorithm 2.1 by $\bar{X}$. By construction,

[3] The center of the cone of semidefinite matrices is given by the minimizer of $-\ln\det(X)$ subject to the constraint that the diagonal of X consists of all ones. In the absence of knowing a self-concordant barrier function for C^* one might approximate the center of C^* by the center of $C_+ := \{X \mid X \succeq 0,\ X \geq 0\}$ which coincides with C^* for $n \leq 4$ and which has the canonical self-concordant barrier function $-\ln\det(X) - \sum_{i<j} \ln X_{i,j}$. Straightforward calculations show that the centers of C_+ (with diagonal all ones) have off-diagonal elements all equal to $(n - 2 + \sqrt{n^2 + 8})/(2n + 2)$. These matrices are also interior to C^*.

Table 1 Results of Algorithm 2.1 for $n = 10$

$n = 10$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
Minimal $\|B - X^0(X^0)^T\|_F$	$5.51 \cdot 10^{-2}$	$4.78 \cdot 10^{-2}$	$6.03 \cdot 10^{-2}$
Maximal $\|B - X^0(X^0)^T\|_F$	$7.05 \cdot 10^{-2}$	$6.47 \cdot 10^{-2}$	$6.39 \cdot 10^{-2}$
Minimal $\|B - X^{end}(X^{end})^T\|_F$	$1.72 \cdot 10^{-6}$	$1.06 \cdot 10^{-12}$	$4.36 \cdot 10^{-13}$
Maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.73 \cdot 10^{-4}$	$1.85 \cdot 10^{-12}$	$8.71 \cdot 10^{-13}$
Minimal number of iterations	11	34	37
Maximal number of iterations	41	39	38
Average running time	78.8 s	5.8 s	3.4 s

Table 2 Results of Algorithm 2.1 for $n = 50$

$n = 50$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
Minimal $\|B - X^0(X^0)^T\|_F$	$6.46 \cdot 10^{-2}$	$3.74 \cdot 10^{-2}$	$1.62 \cdot 10^{-2}$
Maximal $\|B - X^0(X^0)^T\|_F$	$7.94 \cdot 10^{-2}$	$5.02 \cdot 10^{-2}$	$2.28 \cdot 10^{-2}$
Minimal $\|B - X^{end}(X^{end})^T\|_F$	$2.08 \cdot 10^{-8}$	$4.23 \cdot 10^{-14}$	$2.05 \cdot 10^{-14}$
Maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.19 \cdot 10^{-7}$	$7.94 \cdot 10^{-14}$	$3.43 \cdot 10^{-14}$
Minimal number of iterations	27	41	40
Maximal number of iterations	34	44	42
Average running time	764.5 s	123.7 s	34.2 s

Table 3 Results of Algorithm 2.1 for $n = 200$

$n = 200$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
Minimal $\|B - X^0(X^0)^T\|_F$	$1.04 \cdot 10^{-1}$	$6.74 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$
Maximal $\|B - X^0(X^0)^T\|_F$	$1.13 \cdot 10^{-1}$	$7.09 \cdot 10^{-2}$	$3.02 \cdot 10^{-2}$
Minimal $\|B - X^{end}(X^{end})^T\|_F$	$1.10 \cdot 10^{-10}$	$4.05 \cdot 10^{-15}$	$2.41 \cdot 10^{-15}$
Maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.86 \cdot 10^{-10}$	$4.99 \cdot 10^{-15}$	$2.49 \cdot 10^{-15}$
Minimal number of iterations	35	50	57
Maximal number of iterations	41	54	68
Average running time	27609.6 s	1595.4 s	1858.1 s

$\bar{X} \geq 0$ and the matrix $\bar{X}\bar{X}^T$ is closer to B than $X^0(X^0)^T$. While $\bar{X}\bar{X}^T \in C^*$ is in some way "close" to B, it is generally *not* the projection of B onto C^*.

It is somewhat surprising that the final accuracy reached for the "large scale" problems ($n = 200$) is higher than for the smaller problems. This may indicate that large scale random matrices as generated here are "easy"; the "hard" matrices (having high cp-rank?) need to be constructed by other means.

Above examples were also tested by solving the conic subproblems with SeDuMi [14]. For $n = 10$ and $k = 2n$ the running times were 35.2 s in the average, and for $k = n/2$ they were 38.2 s in the average. (The time for generating the matrices that define the SeDuMi subproblems (1.5 s) was negligible for $n = 10$). For $n = 50$ the SeDuMi-based approach took about 5,290 s in the average. (Here, the time for generating the matrices that define the SeDuMi subproblems was significant (another 3,620 s).) For $n \geq 100$ SeDuMi could not be started on this computer. Thus, for small n, SeDuMi is competitive (and in case of $k = n/2$ even slightly faster than the apd approach), and, as expected, for moderate values of n, the apd-method is faster.

These examples showed another unexpected feature. The corrections ΔX computed by SeDuMi solved the subproblems (3) to much higher accuracy than the corrections ΔX computed by the apd method. On the other hand, the norm of the corrections ΔX generated by the apd method was smaller. For $n = 10$ and $k = 2n$, the final accuracy achieved by Algorithm 2.1 with SeDuMi was $6.2 \cdot 10^{-10}$ compared to $6.4 \cdot 10^{-13}$ with apd. (This indicates that the subproblems should be formulated differently when using SeDuMi, for example minimizing a weighted average of the norm of the residual of the equality constraints and of $\|\Delta X\|_F$.) For $n = 10$ and $k = n/2$, the final accuracy achieved by Algorithm 2.1 with SeDuMi was $5.1 \cdot 10^{-8}$ and thus significantly better than the apd-approach. In the final iterations the apd-method was not able to render sufficiently accurate solutions of the subproblems for these more difficult examples. Improvements of the apd-method are the subject of ongoing research and will also influence the results of Algorithm 2.1.

3 Generating a starting point

3.1 The diagonal of B

Algorithm 2.1 considers the problem whether a given symmetric matrix B is in C^*. When B has a negative eigenvalue or a negative matrix entry, then trivially $B \notin C^*$. Hence, we may assume $B \succeq 0$ and $B \geq 0$. If the positive semidefinite matrix B has a zero diagonal element then the corresponding row and column of B is zero and the task of finding $X \geq 0$ with $XX^T = B$ can be reduced to a smaller dimensional problem. For the generation of a starting point in Sect. 3.2 we therefore assume that B has strictly positive diagonal entries.

Let D be the positive definite diagonal matrix such that D^{-2} coincides with the diagonal of B. Given a nonnegative factorization $DBD = \tilde{X}\tilde{X}^T$, it is trivial to recover the nonnegative factorization $B = (D^{-1}\tilde{X})(D^{-1}\tilde{X})^T$. Hence when defining a starting point X^0 such that $X^0(X^0)^T \approx B$ we may rescale $B := DBD$ to have a diagonal of all ones. When $B \in C^*$ this implies that $B_{i,j} \in [0, 1]$ for all i, j.

3.2 Criteria for the starting point

In this section a starting point $X^0 \in I\!R^{n\times p}$, $X^0 \geq 0$ is defined such that $S^0 := X^0(X^0)^T \approx B$ and such that S^0 lies near the "center" of C^*.

The choice of $X^0 \geq 0$ with $X^0(X^0)^T = S^0$ is far from unique as for a given matrix $S = S^0 \in C^*$ of cp-rank $\leq p$ the set

$$\Xi_p(S) := \{X \in I\!R^{n\times p} \mid X \geq 0,\ XX^T = S\} \tag{10}$$

contains more than one element (unless $p = 1$). As shown in [7], for any S in the interior $(C^*)^\circ$ of C^*, there exists a representation $S = XX^T$ satisfying

$$X = [X^1, X^2], \quad 0 < X^1 \in I\!R^{n\times n}, \quad \text{and} \quad X^1(X^1)^T \succ 0. \tag{11}$$

On the other hand, even when $S \in (C^*)^\circ$ there may also exist representations XX^T of S that violate (11). For example, $S = I + (n+2)E$ has the representations

$$S = XX^T = \hat{X}\hat{X}^T \quad \text{with } X = [E+I, 0] \quad \text{and} \quad \hat{X} = [I, \sqrt{n+2}e].$$

The representation $\hat{X}\hat{X}^T$ not only violates (11), but, as will be detailed next, it is also less suitable for the computation of corrections $\Delta\hat{X}$:

Let us define the perturbation ΔS with entries $\Delta S_{i,j} = 0$ for all i, j except from $\Delta S_{1,2} = \Delta S_{2,1} = -1$. We consider corrections $\Delta\hat{X}$ and ΔX such that $\hat{X} + \Delta\hat{X} \geq 0$ and $X + \Delta X \geq 0$ satisfy the linearized equations

$$\Delta\hat{X}\hat{X}^T + \hat{X}\Delta\hat{X}^T = \Delta S \quad \text{and} \quad \Delta X X^T + X\Delta X^T = \Delta S. \tag{12}$$

Straightforward calculations show that the minimum norm solution $\Delta\hat{X}$ of (12) has a norm of about $\sqrt{2n-4}$. On the other hand, for any ΔS of norm $\sqrt{2}$ (including the above perturbation ΔS), the minimum norm solution ΔX of (12) is bounded by $\sqrt{2}$. In this example, the zero entries in $\hat{X}$ restrict the choice of the corrections $\Delta\hat{X}$.

As the linearization error $\Delta X \Delta X^T$ increases with $\|\Delta X\|$ the representation XX^T appears to be more suitable as a starting point for linearized corrections $X \to X + \Delta X$ than the representation $\hat{X}\hat{X}^T$. We summarize the crucial points for determining an initial matrix X^0:

- When X^0 has two or more identical columns, these columns will remain identical throughout the algorithm. (This will only increase computation time.) Below we generate nonnegative columns that have pairwise a "large angle" to each other.
- The matrix X^0 below is generated such that it contains a strictly positive $n \times n$ submatrix whose smallest singular value is "large". By (11), this guarantees that $S^0 \in (C^*)^\circ$.
- Of course, the algorithm is expected to profit from a "warm start". On the other hand, when using a warm start at the boundary of C^*, e.g. with a singular matrix S^0, the algorithm may get stuck at the first iteration. In this work we give priority to a "central" starting point over a good initial approximation $X^0(X^0)^T \approx B$.

3.3 Two specific starting points

We consider two possible choices of p:

1. When n is large, a choice of $p \geq n^2/4$ may be infeasible due to limitations in storage and computation time. In this case it may suffice to find an approximation $XX^T \approx B$ that improves the initial decomposition $X^0(X^0)^T \approx B$. We then choose $n < p \leq 2n$ and a starting point X^0 is evaluated by the following steps:
   By symmetric permutations the columns of B are reordered in increasing norm, $\tilde{B} := \Pi^T B \Pi$. Then, a Cholesky factorization $\tilde{B} = LL^T$ is computed (when $\tilde{B}$ is singular, L has fewer than n columns). If $L \geq 0$ stop ($B \in C^*$); else project L onto the set of nonnegative matrices. Finally, the rows of L are permuted back; $L := \Pi L$. Let $e \in \mathbb{R}^n$ be the vector of all ones and e_i be the i-th unit vector for $1 \leq i \leq n$. The first n columns of X^0 are set to $\frac{1}{2n}e + \frac{1}{\sqrt{n}}e_i$ $(1 \leq i \leq n)$. The remaining $p - n$ columns are set to the first $p - n$ columns of L. (Reduce p when L has fewer than $p - n$ nonzero columns.) Let $\hat{D}$ be the diagonal of $X^0(X^0)^T$. To match the diagonal of B and $X^0(X^0)^T$ we set $X^0 := \hat{D}^{-1/2}X^0$. (This is the approach used for the examples in Sect. 2.6.)
2. Second, $p = n(n+1)/2$. This second option is feasible only for small sizes of n, say $n \leq 50$. In this case, the following procedure generates a matrix $X^0(X^0)^T$ in the "center" of C^*:

As above, the first n columns of X^0 are set to $\frac{1}{2n}e + \frac{1}{\sqrt{n}}e_i$ $(1 \leq i \leq n)$. The remaining $n(n-1)/2$ columns are set to $\frac{1}{n^2}e + \frac{1}{n}(e_i + e_j)$ $(1 \leq i < j \leq n)$. It is easy to see that $X^0(X^0)^T = \lambda ee^T + \rho I$ where I is the identity matrix and λ, ρ are positive scalars. When changing the factors $\frac{1}{2n}, \frac{1}{\sqrt{n}}, \frac{1}{n^2}, \frac{1}{n}$ to other positive values, the numbers λ, ρ will change. $X^0(X^0)^T$ is in the interior of C^* if, and only if, $\lambda > 0$ and $\rho > 0$. Normalizing $X^0(X^0)^T$ to diagonal of all ones is simply achieved by setting $X^0 := \frac{1}{\sqrt{\lambda+\rho}}X^0$.

The above construction not only generates a "central" matrix S^0 but also a "central" $X^0 \in \Xi(S^0)$ supporting the goal of defining a "central" path $[S^0, B)$. If a "good" factorization $B \approx VV^T$ with $V \in I\!R_+^{n\times k}$ is known, this initial point may of course be extended to $X^0 \leftarrow [\sqrt{\lambda}V, \sqrt{1-\lambda}X^0]$ for some $\lambda \in (0, 1)$ – increasing the value p to $k + n(n+1)/2$.

Whether or not the matrix B is scaled back $B := D^{-1}BD^{-1}$ (and likewise $X^0 := D^{-1}X^0$) before starting Algorithm 2.1 in the next section depends on the norm in which we would like to measure the distance between XX^T and B.

4 A regularization step

As outlined before, when S has full rank and X is strictly positive with $XX^T = S$ the linear Eq. 2 have a feasible solution. Given a matrix $S = S^k$ at iteration k, this suggests to search—among all matrices X with $XX^T = S$—for a strictly positive matrix X before computing the correction step (*Step 3.* of Algorithm 2.1). Below we present a heuristic for generating matrices $X \in \Xi_p(S)$ (see (10)) whose smallest entries are as large as possible.

Let $S = X^k(X^k)^T$ denote a certain iterate of Algorithm 2.1. The goal of this section is to compute a "central" element $\bar{X}$ of $\Xi_p(S)$ in the sense that $\bar{X} - \bar{\rho}E \geq 0$ for a large value of $\bar{\rho}$. Here, E is the 'all-ones-matrix'. (When $\bar{\rho} > 0$ and ΔX is given arbitrarily, this allows a correction $\bar{X} \mapsto \bar{X} + \epsilon\Delta X$ for some $\epsilon > 0$ without violating the nonnegativity constraints.) The regularization step can be applied after each iteration of Algorithm 2.1 replacing X^k with a "more central" matrix $\bar{X}^k$.

The following proposition is used to generate such a "central element".

Proposition 4.1 *Given $S \succeq 0$ with distinct eigenvalues $\lambda_i > \lambda_{i+1}$ for $1 \leq i \leq n-1$ and $X, \bar{X}$ with $XX^T = S = \bar{X}\bar{X}^T$ then $\bar{X} = X\hat{V}$ for some unitary matrix $\hat{V}$.*

Proof Let $X = U\Sigma V$ and $\bar{X} = \bar{U}\bar{\Sigma}\bar{V}$ be the singular value decompositions of X and $\bar{X}$ where the singular values $\Sigma_{i,i} = \bar{\Sigma}_{i,i} = \sqrt{\lambda_i}$ are arranged in decreasing order. Comparing XX^T and $\bar{X}\bar{X}^T$ one obtains

$$U\Sigma\Sigma^T U^T = \bar{U}\bar{\Sigma}\bar{\Sigma}^T\bar{U}^T,$$

i.e. $\Sigma\Sigma^T = U^T\bar{U}\bar{\Sigma}\bar{\Sigma}^T\bar{U}^T U$. As $\Sigma\Sigma^T = \bar{\Sigma}\bar{\Sigma}^T$ is a diagonal matrix with strictly decreasing diagonal entries and $U^T\bar{U}$ is unitary, it follows that $U^T\bar{U} = I$, i.e. $U = \bar{U}$. Defining $\hat{V} := V^T\bar{V}$ the claim of the proposition follows. □

Remark 4.1 When the eigenvalues of S are not pairwise distinct there might be additional degrees of freedom in the selection of X and $\bar{X}$. This possibility is not exploited in this paper.

Proposition 4.1 shall be used to change a given matrix $X \in \Xi(S)$ to a slightly 'more central' matrix $\bar{X}$. The change will be based on a "linearization" of the matrix $\hat{V}$, so that the equality $XX^T \approx \bar{X}\bar{X}^T$ only holds approximately.

By $\$$ we always denote a *skew symmetric* matrix, $\$ = -\$^T$. For small $\|\hat{V} - I\|$ it follows that there exists a skew symmetric matrix $\$$ such that

$$\hat{V} = exp(\$) = I + \$ + O(\|\$\|^2).$$

(This equation defines the "linearization" referred to above.) Given a matrix $X \in \Xi(S)$ we search for a small correction of the form

$$X \mapsto \bar{X} := X(I + \epsilon\$) \tag{13}$$

such that $\bar{X} \geq \bar{\rho} E$ for a large value of $\bar{\rho}$. To this end the matrix $\$$ is determined by the linear program

$$\text{maximize } \ \rho \mid X(I + \$) \geq \rho E$$

which can be written in the dual form

$$\text{maximize } \ \rho \mid \rho E - X\$ \leq X. \tag{14}$$

Whenever the optimal solution to (14) has an optimal value that is larger than $\min_{i,j} X_{i,j}$ an update of the form (13) with $\epsilon \in (0, 1)$ will increase the lower bound $\min_{i,j} \bar{X}_{i,j}$—at the expense of a second-order perturbation to XX^T.

When $\$$ is a $p \times p$-matrix, the linear program (14) has about $p^2/2$ unknowns and np inequalities of a special structure. Here, p may be a rather large number (typically one would choose $p \leq n^2/2$), and for $p > 2n$ the program has more variables than inequalities. It is easy (we omit the details) to find primal and dual feasible points for this program. Thus, the program has a finite optimal solution but (at least for $p > 2$) it is highly degenerate. Again, the linear operator defining the constraints is not given by a matrix, and the structure of this linear program is not suitable for interior-point methods.

The solution of (14) can also be computed by the apd algorithm, and as in Sect. 2, the accuracy of the solution of the subproblems can be adjusted according to the linearization error and the distance of XX^T to B.

4.1 Standard form of the apd-algorithm

Let the mapping $\mathcal{A}^*$ be given by $\mathcal{A}^*(\$) = X\$$. $\mathcal{A}^*$ maps the space of skew symmetric $p \times p$-matrices to $I\!R^{n\times p}$. Its adjoint is given by

$$\mathcal{A}(Z) = \frac{1}{2}(X^T Z - Z^T X)$$

for $Z \in I\!R^{n\times p}$. With this notation, the primal of (14) is given by

$$\text{minimize } \ X \bullet Z \mid E \bullet Z = 1, \ \ \mathcal{A}(Z) = 0, \ \ Z \geq 0.$$

Note that $Z^0 := X/(E \bullet X)$ is feasible for the primal problem.

To apply the apd-algorithm [9] to this LP we denote

$$\mathcal{L} := \{Z \mid E \bullet Z = 0, \ \ \mathcal{A}(Z) = 0\}.$$

The primal problem can thus be written as

$$\text{minimize } \ X \bullet Z \mid Z \in (\mathcal{L} + Z^0) \cap I\!R^{n\times p}_+.$$

Here, the iterate X is given data (it is the goal of this LP to increase the minimum entry of X) and Z is the dual variable.

We recall that the apd-algorithm is based on the availability of cheap projections onto $\mathcal{L}$. These will be discussed next. (The authors were not able to provide equally cheap solutions for the linear systems that arise in interior-point approaches for this problem.)

The KKT conditions for the projection ΔZ of a matrix Z onto $\mathcal{L}$ can be written as follows: There exists a $\rho \in I\!R$ and a skew symmetric \$ such that

$$\rho E \bullet E + E \bullet \mathcal{A}^*(\$) = E \bullet Z$$
$$\rho \mathcal{A}(E) + \mathcal{A}(\mathcal{A}^*(\$)) = \mathcal{A}(Z).$$

In the sequel the brackets as in $\mathcal{A}(E)$ will be omitted and we simply write $\mathcal{A}E$. The pseudoinverse of a linear operator M is denoted by M^+. With this notation, the solution of this system can be obtained via

$$\rho = \frac{Z \bullet (E - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^+ \mathcal{A}E}{E \bullet (E - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^+ \mathcal{A}E}$$
$$\$ = (\mathcal{A}\mathcal{A}^*)^+ \mathcal{A}(Z - \rho E).$$

Note that E is not contained in the range of $\mathcal{A}^*$ and hence, ρ is well-defined. (If E was contained in the range of $\mathcal{A}^*$ the linear program (14) would be unbounded.)

We briefly discuss the least squares solution of the system $\mathcal{A}\mathcal{A}^*\$ \approx R$ for some given skew symmetric right hand side $R \in I\!R^{p\times p}$. (This least squares solution coincides with $\$ = (\mathcal{A}\mathcal{A}^*)^+ R$.)

Observe that $\mathcal{A}\mathcal{A}^*\$ = \frac{1}{2}(X^T X\$ + \$X^T X)$. We obtain the equation

$$2\mathcal{A}\mathcal{A}^*\$ = X^T X\$ + \$X^T X \approx 2R$$

for the unknown matrix \$. We assume that the singular value decomposition of X is given, $X = U\Sigma V$ with unitary matrices U and V of suitable dimensions. Using the singular value decomposition and setting $\tilde{\$} = V\$V^T$ this is equivalent to

$$\Sigma^T \Sigma \tilde{\$} + \tilde{\$}\Sigma^T \Sigma \approx 2\tilde{R} := 2VRV^T.$$

The matrix $\tilde{\$}$ is skew symmetric as well, and the above unitary transformations do not change the least squares solution. Here, $\Sigma^T \Sigma$ is a $p \times p$ diagonal matrix, only the leading n diagonal entries of it being nonzero (when X has maximum rank, else there are $r < n$ nonzero entries).

Solving this system for $\tilde{\$}$ in a least squares sense is trivial, yielding the desired solution $\$ = V^T \tilde{\$} V$.

Above computations require about $O(p^3)$ operations. Note that $\mathcal{A}\mathcal{A}^*$ maps the skew symmetric $p \times p$ matrices into themselves; the inversion of a general map $I\!R^{p\times p} \to I\!R^{p\times p}$ may take $O(p^6)$ operations.

4.2 Recovering the primal variable

If problem (14) is solved by the apd-method, the last step of the algorithm can be chosen as the projection onto the affine hull of the primal dual feasible solutions. We obtain a primal dual solution in the apd-format satisfying all equality constraints. The dual solution N is a matrix in $I\!R^{n\times n}$ such that there exist variables \$ and ρ with

$$\rho E - X\$ = R := X - N.$$

(Ideally, when also the primal dual inequalities are satisfied then $N \in I\!R^{n\times n}_+$, but due to the last projection this cannot be guaranteed.) We are then interested in the values of \$ and ρ.

Specifically, we need to solve a system of the form

$$\rho E - U\Sigma V\$ = R$$

where $U\Sigma V$ is the singular value decomposition of X. (For general R this system may not have a solution, but by our assumption that the apd method terminates with a projection on the primal dual feasible equations a solution must exist.) Setting $\tilde{E} := U^T E V^T$, $\tilde{\$} := V\$V^T$, and $\tilde{R} := U^T R V^T$, this system is equivalent to

$$\rho\tilde{E} - \Sigma\tilde{\$} = \tilde{R}.$$

Let $\Sigma = [D, 0]$. By the assumption $XX^T \in (C^*)^\circ$ it follows that X must have full rank and thus D is an $n \times n$ positive definite diagonal matrix. We obtain

$$\rho D^{-1}\tilde{E} - [I, 0]\tilde{\$} = D^{-1}\tilde{R}.$$

Note that $\tilde{\$}$ is skew symmetric and thus has a zero diagonal. Hence we may determine ρ such that $D^{-1}(\tilde{R} - \rho\tilde{E})$ has a zero diagonal. (In the presence of rounding errors a least squares solution ρ may be used.) Once ρ is given, the computation of $\$$ is straightforward.

4.3 The effect of the regularization step

For $B = \begin{pmatrix} lI & E \\ E & lI \end{pmatrix}$ the results of Algorithm 2.1 without regularization step were disappointing. As pointed out, this matrix has cp-rank $n^2/4$. To have any chance to prove complete positivity for this choice of B we applied Algorithm 2.1 with the expensive choice $p = n(n+1)/2$ and starting point 2. in Sect. 3.3. The example was tested first with Algorithm 2.1 without regularization. Due to slow convergence we stopped the algorithm after 260 iterations with a residual of $1.56 \cdot 10^{-2}$. For this example we also tested the regularized approach, allowing 10 regularization steps after each iteration. (Each regularization step solves a linear program with $p(p-1)/2$ variables and is thus very expensive.) After 65 iterations the regularized approach obtained an accuracy of $4.85 \cdot 10^{-5}$.

Since the results of Algorithm 2.1 for $n = 10$ and $k = n/2$ were disappointing as well, we also tested the regularized approach for these examples. The corresponding results for the regularized approach in comparison with the approach without regularization are given in Table 4.

The approach without regularization was stopped whenever the algorithm stagnated or the iteration number exceeded twice the iteration number of the regularized approach. The

Table 4 Results of Algorithm 2.1 with/without regularization for $n = 10$

$n = 10$	With regularization	Without regularization
Minimal $\|B - X^0(X^0)^T\|_F$	$5.62 \cdot 10^{-2}$	
Maximal $\|B - X^0(X^0)^T\|_F$	$1.06 \cdot 10^{-1}$	
Minimal $\|B - X^{end}(X^{end})^T\|_F$	$4.38 \cdot 10^{-8}$	$3.44 \cdot 10^{-6}$
Maximal $\|B - X^{end}(X^{end})^T\|_F$	$3.25 \cdot 10^{-7}$	$1.04 \cdot 10^{-4}$
Minimal number of iterations	17	10
Maximal number of iterations	29	30
Average time (sec)	1, 490	79

algorithm with regularization performs better when the constant ϵ in the regularization step is chosen in dependence of the distance of the current iterate $X^k(X^k)^T$ to B with smaller values of $\epsilon > 0$ when $X^k(X^k)^T$ is close to B.

The aim of this experiment was to test whether the regularization step really does improve the overall convergence. While the improved convergence behavior of the regularized approach is indeed encouraging, the running times leave room for improvement—with 10 regularization steps after each iteration the regularized approach took about 20 times longer. Let

$$X(\tilde{\epsilon}) := X\left(I + \tilde{\epsilon}\$ + \frac{1}{2}\tilde{\epsilon}^2\$^2\right), \qquad X(\tilde{\epsilon}) := X\left(I + \tilde{\epsilon}\$ + \frac{1}{2}\tilde{\epsilon}^2\$^2 + \frac{1}{6}\tilde{\epsilon}^3\$^3\right),$$
$$\text{or} \quad X(\tilde{\epsilon}) := X\exp(\tilde{\epsilon}\$).$$

In the first two cases the error term $X(\tilde{\epsilon}) - X\exp(\tilde{\epsilon}\$)$ can be estimated by $E(\tilde{\epsilon}) := \frac{1}{6}\tilde{\epsilon}^3X\$^3$ or $E(\tilde{\epsilon}) := \frac{1}{24}\tilde{\epsilon}^4X\$^4$. Instead of 10 short steps of the form (13) one or two steps with a line search along $X(\tilde{\epsilon})$ minimizing some weighted average of "$-\min_{i,j} X(\tilde{\epsilon})_{i,j}$" and $\|E(\tilde{\epsilon})\|$ might be sufficient. We did not explore this or other options to reduce the computation time.

5 Conclusion

In this paper, the quadratic factorization heuristic—proposed in a different context in [4]—is used for the generation of a certificate of complete positivity of a given matrix B; or for completely positive completion problems. The algorithm generates iterates that are determined by approximate solutions of certain subproblems. These subproblems can be reformulated as second-order cone programs. Due to a linearization error, the exact solution of the subproblems does not generate the desired certificate but merely determines a step towards the next iterate. Because of this linearization error the implementation solves the subproblems only up to a precision of same the magnitude as the linearization error. For such approximate solutions the apd-method [9,13] is well suited. Numerical results show a very promising convergence behavior of the algorithm for matrices B in the interior of C^* and of low cp-rank. The convergence slows down significantly, when B is on the boundary of C^* or when B has a large cp-rank. To accelerate the algorithm for this case we propose a novel regularization step after each iteration aiming at making all matrix entries of the current factor X^k as large as possible without changing the product $X^k(X^k)^T$. Numerical examples illustrate the positive effect of this regularization. It was possible to test this regularization—and to establish its positive effect on the overall algorithm—for small size problems. Due to limits in computation time, the application to large problems remains a topic of future research.

Acknowledgments The authors would like to thank two anonymous referees whose constructive comments were of great help in improving the paper.

References

1. Berman, A., Rothblum, U.: A note on the computation of the CP-rank. Linear Algebra Appl. **419**, 1–7 (2006)
2. Berman, A., Shaked-Monderer, N.: Completely Positive Matrices. World Scientific, Singapore (2003)
3. Bomze, I.M., Dür, M., de Klerk, E., Roos, C., Quist, A.J., Terlaky, T.: On copositive programming and standard quadratic optimization problems. J. Glob. Optim. **18**, 301–320 (2000)

4. Bomze, I.M., Jarre, F., Rendl, F.: A quadratic factorization heuristic for copositive programs, Preprint, in preparation (2007)
5. Burer, S.: On the copositive representation of binary and continuous nonconvex quadratic programs. Preprint, University of Iowa (2007), available at http://www.optimization-online.org/DB_HTML/2006/10/1501.html
6. de Klerk, E., Pasechnik, D.V.: A linear programming reformulation of the standard quadratic optimization problem. J. Glob. Optim. **37**, 75–84 (2007)
7. Dür, M., Still, G.: Interior points of the completely positive cone. Electron. J. Linear Algebra **17**, 48–53 (2008) (ISSN 1081-3810)
8. Hall, M. Jr., Newman, M.: Copositive and completely positive quadratic forms. Proc. Camb. Philos. Soc. **59**, 329–339 (1963)
9. Jarre, F., Rendl, F.: An augmented primal-dual method for linear conic programs. SIAM J. Optim. **19**(2), 808–823 (2008). doi:10.1137/070687128
10. Maxfield, J.E., Minc, H.: On the matrix equation $X'X = A$. Proc. Edinb. Math. Soc. **13**, 125–129 (1962/1963)
11. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and linear programming. Math. Program. **39**, 117–129 (1987)
12. Nesterov, Y., Nemirovskii, A.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, Philadelphia (1994)
13. Schmallowsky, K.: On the regularity of second order cone programs and an application to solving large scale problems. Math. Methods Oper. Res. (2008) (to appear)
14. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11–12**, 625–653 (1999)