



Suliman Al-Homidan

Second-order cone and semidefinite methods for the bisymmetric matrix approximation problem

Received: 30 January 2022 / Accepted: 12 June 2022 / Published online: 23 August 2022
© The Author(s) 2022

Abstract Approximating the closest positive semi-definite bisymmetric matrix using the Frobenius norm to a data matrix is important in many engineering applications, communication theory and quantum physics. In this paper, we will use the interior point method to solve this problem. The problem will be reformulated into various forms, in the beginning as a semi-definite programming problem and later, into the form of a mixed semidefinite and second-order cone optimization problem. Numerical results comparing the efficiency of these methods with the alternating projection algorithm will be reported.

Mathematics Subject Classification 49M15 · 52A27 · 65F30 · 90C90

1 Introduction

Bisymmetric matrices, that is, symmetric and persymmetric matrices, have been extensively discussed since 1931, which are very effective in engineering problems and have practical applications in information theory, communication theory, quantum physics, linear system theory, numerical analysis and statistics. Some results for the inverse problem of bisymmetric matrices have been obtained in [1, 11, 13]. The least squares bisymmetric and partially bisymmetric solutions have been studied by [9, 19]. Other problems related to bisymmetric matrices have been studied by [7, 14, 21]. The persymmetric Hankel and symmetric Toeplitz matrices are special forms of bisymmetric matrices that usually appear in different applications, such as vibration in structures, matching filters and estimates of shaping [4, 6].

In some of these application areas, it is required to evaluate the nearest positive semi-definite bisymmetric matrix to a given data matrix. The required matrix has no restriction on its rank. An existence theorem for the problem solutions is given in [20], and the general expression of the solutions is derived using Newton method. Similar problems containing structured covariance estimation were studied in [2–4].

Interior point algorithms can solve semi-definite programming problems efficiently in polynomial time [5]. This motivates us to construct the problem as mixed second-order cone and semi-definite programming problem. The constraints in this problem are constraints over the second-order cone and the positive semi-definite cones. The software SDPT3 by Toh et al. [17] is an efficient software for implementation of primal-dual path-following methods that will be used to solve the problem.

There are many similar semi-definite programming problems, for example Todd [16] formulated the optimization of the spectral norm of a special matrix as a semi-definite programming problem. Later, some other problems were compiled as mixed semi-definite and second-order cone optimization problems [15]. The special structure of our problem is an advantage, however, none of the above formulations utilized it.

S. Al-Homidan (✉)

Department of Mathematics and Statistics, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
E-mail: homidan@kfupm.edu.sa



To take advantage of bisymmetric structure, an isometry operator **bvec** is introduced. The isometry operator, **bvec** takes $n \times n$ bisymmetric matrices into r vectors, where r given (1.3) and it is much smaller than n^2 . This operator will give our methods an advantage over the other methods.

Some notations are introduced now that will be used throughout the paper. The set of all $n \times n$ real symmetric matrices will be denoted by \mathcal{S}_n . The cone of all $n \times n$ real symmetric positive semi-definite matrices will be denoted by \mathcal{S}_n^+ , where

$$\mathcal{S}_n^+ = \{A : A \in \mathbb{R}^{n \times n}, \text{ and } z^T A z \geq 0, \forall z \in \mathbb{R}^n\} \quad (1.1)$$

is a convex cone of dimension $n(n+1)/2$. \mathcal{Q}_h is a second-order cone of dimension h , and is defined as

$$\mathcal{Q}_h = \{y \in \mathbb{R}^h : \|y_{2:h}\|_2 \leq y_1\},$$

where $\|\cdot\|_2$ stands for the Euclidean distance norm defined as $\|y\|_2 = \sqrt{\sum_{i=1}^n y_i^2}$, $\forall y \in \mathbb{R}^n$ and $x_{2:h} = [x_2, x_3, \dots, x_h]^T$. The set of all $n \times n$ real bisymmetric matrices will be denoted by \mathcal{B}_n , where

$$\mathcal{B}_n = \{B(b) : B(b) \in \mathbb{R}^{n \times n}, B(b) \text{ is bisymmetric}\}, \quad (1.2)$$

which is a subspace of dimension r . In the following, the structure of an $n \times n$ real bisymmetric matrix $B(b)$:

$$B(b) = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \\ b_2 & b_{n+1} & \cdots & b_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{2n-2} & \cdots & b_2 \\ b_n & b_{n-1} & \cdots & b_1 \end{bmatrix}, \quad b \in \mathbb{R}^r, \quad (1.3)$$

where $r = mn - k$, $m = n/2$ and $k = n(n-2)/4$ if n is even and $m = (n+1)/2$ and $k = (n-1)(n+1)/4$ if n is odd. It is obvious that $\mathcal{B}_n \subset \mathcal{S}_n$. The norm defined on \mathcal{S}_n is the Frobenius norm expressed as follows:

$$\|W\|_F = \sqrt{W \bullet W} = \|\mathbf{vec}^T(W) \mathbf{vec}(W)\|_2, \quad \forall W \in \mathcal{S}_n. \quad (1.4)$$

Here $W \bullet W = \text{trace}(W \cdot W) = \sum_{i,j} W_{i,j}^2$, $\mathbf{vec}(W)$ means vectorization operator founded by putting all columns of the matrix W on top of each other as one column and \mathbf{vec}^T is the transpose of \mathbf{vec} . We denote the partial ordering on \mathcal{S}_n^+ and \mathcal{Q}_h on \mathcal{S}_n and \mathbb{R}^h , respectively, by \succeq and $\succeq_{\mathcal{Q}}$. That is,

$$W \succeq P \Leftrightarrow W - P \in \mathcal{S}_n^+, \quad \forall W, P \in \mathcal{S}_n$$

and

$$w \succeq_{\mathcal{Q}} p \Leftrightarrow w - p \in \mathcal{Q}_h, \quad \forall w, p \in \mathbb{R}^k,$$

where $y \geq 0$ for a vector $y \in \mathbb{R}^n$ stands for each component of y being nonnegative. We denote the zero and identity matrices by 0 and I , respectively.

We can now describe our problem in mathematical notation as follows: Given a data matrix $G \in \mathbb{R}^{n \times n}$, find the closest positive semi-definite bisymmetric matrix $B(b)$ to G such that $\|G - B(b)\|_F$ is minimal. Thus, we have the following minimization problem:

$$\begin{aligned} & \text{minimize } \|G - B(b)\|_F \\ & \text{subject to } B(b) \in \mathcal{B}_n, \quad B(b) \succeq 0. \end{aligned} \quad (1.5)$$

In Sect. 2, we describe briefly the alternating projection method. Although the rate of convergence is slow, the modified alternating projection method converges globally to the optimal solution. We can compare the results achieved by alternating projection method with our methods in Sects. 3 and 4, since alternating projection method provides us with an accurate solutions. A brief description of semi-definite and second-order cone minimization problems alongside the interior point formulas of problem (1.5) in the formation of the relevant class will be given in Sects. 3 and 4, respectively. The performance of these primal-dual path-following methods against the alternating projection method is shown in numerical results Sect. 5.



2 The alternating projection method

The algorithm of this section is obtained from modified alternating projection method that are originally proposed by von Neumann [18] for finding the minimum distance from a given fixed point to an intersection of convex sets.

When applying the projection method to approximate a bisymmetric matrices, it is convenient to use the Frobenius norm, as expressed (1.4). To apply projection method, the projection maps $P_S(\cdot)$ and $P_B(\cdot)$ are needed. These projections are the maps from $K = \{G : G \in \mathbb{R}^{n \times n}\}$ on to S_n^+ and B_n . The projection mapping $P_S(G)$ on to S_n^+ is given by the formula [10]

$$P_S(G) = U\Lambda^+U^T, \tag{2.1}$$

where

$$\Lambda^+ = \begin{bmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

and $\Lambda_r = \text{diag} [\lambda_1, \lambda_2, \dots, \lambda_r]$ is the diagonal matrix consisting from the nonnegative eigenvalues of Λ .

The mapping $P_B(G)$ formula onto B_n is now given by

$$P_B(G) = \text{Bis}(b_1, b_2, \dots, b_r), \tag{2.2}$$

where b_p in the (i, j) position of the matrix (1.3) is given by

$$b_p = (G_{i,j} + G_{j,i} + G_{(n-i+1,n-j+1)} + G_{(n-j+1,n-i+1)})/4.$$

The single valued projection mappings $P_S(G)$ and $P_B(G)$ given in (2.1) and (2.2) can now be used to execute the Dykstra algorithm. A data matrix $G \in \mathbb{R}^{n \times n}$ is given, then the method is initialized by $G^{(0)} = G$. The iteration algorithm is given now by:

$$G^{(k+1)} = G^{(k)} + (P_B(P_S(G^{(k)}))) - P_S(G^{(k)}). \tag{2.3}$$

Both sequences $\{P_S(G^{(k)})\}$ and $\{P_B(P_S(G^{(k)}))\}$ produced by (2.3) converge globally to the optimal solution B^* of (1.5), [8].

3 Semidefinite programming approach

The primal standard form for semi-definite programming (SDP) problem is given by:

$$\begin{aligned} (P) \quad & \text{minimize}_X \quad D \bullet X \\ & \text{subject to} \quad A_i \bullet X = b_i, \quad i = 1, 2, \dots, m, \quad X \succeq 0, \end{aligned} \tag{3.1}$$

where all $A_i, D \in S_n, b \in \mathbb{R}^m$ are given and the variable is $X \in S_n$. This minimization problem (3.1) is a convex minimization problem since its constraint and objective are convex. The dual problem of (3.1) is

$$\begin{aligned} (D) \quad & \text{maximize}_y \quad b^T y \\ & \text{subject to} \quad \sum_{i=1}^m y_i A_i \preceq D, \end{aligned} \tag{3.2}$$

where the variable is $y \in \mathbb{R}^m$. Many problems are special cases of problems (3.1) and (3.2) and there are many applications, in particular (1.5). The following lemma is useful in writing (1.5) in the form of (3.2):

Lemma 3.1 (Schur Complement) *If*

$$N = \begin{bmatrix} B & D \\ D^T & C \end{bmatrix}$$

where $B \in S_n^+$ is nonsingular matrix and $C \in S_n$, then the matrix N is positive (semi)definite if and only if the matrix $C - D^T B^{-1} D$ is positive (semi)definite.

This matrix $C - D^T B^{-1} D$ is called the Schur complement of B in N .

3.1 Formulation I (SDV)

Letting $\|G - B(b)\|_F^2 \leq t$, where t is a real nonnegative scalar and observing that:

$$\|G - B(b)\|_F^2 = \mathbf{vec}^T(G - B(b))\mathbf{vec}(G - B(b)),$$

we have:

$$\begin{aligned} & \mathbf{vec}^T(G - B(b))\mathbf{vec}(G - B(b)) \leq t \\ \Leftrightarrow & t - \mathbf{vec}^T(G - B(b))\mathbf{vec}(G - B(b)) \geq 0 \\ \Leftrightarrow & \begin{bmatrix} I & \mathbf{vec}(G - B(b)) \\ \mathbf{vec}^T(G - B(b)) & t \end{bmatrix} \succeq 0. \end{aligned}$$

We can deduce from Lemma 3.1 the equivalency in the above inequalities. Therefore, we can write (1.5) as follows

$$\begin{aligned} (SDV) \quad & \text{minimize } t, \\ & \text{subject to } \begin{bmatrix} t & 0 & 0 \\ 0 & B(b) & 0 \\ 0 & 0 & U \end{bmatrix} \succeq 0, \end{aligned} \tag{3.3}$$

where

$$U = \begin{bmatrix} I & \mathbf{vec}(G - B(b)) \\ \mathbf{vec}^T(G - B(b)) & t \end{bmatrix},$$

this is an SDP problem in the dual form (3.2) and the dimensions of this problem are $r + 1$ (see (1.3) number of variables) and $n^2 + n + 2$ (size of the matrices).

3.2 Formulation II (SDB)

Now, the SDP problem (3.3) is very huge even for a modest data matrix G . For example, a 30×30 matrix G will make the problem grow to a problem with dimensions 240 and 932, thus it is not efficient to solve (1.5) using formulation (3.3). Furthermore, the structure of the matrix $B(b)$ being bisymmetric is not utilized. An alternative approach is to develop an SDP problem with acceptable dimensions and utilizes the bisymmetric structure of $B(b)$. This can now be achieved by means of the following isometry operator.

Definition 3.2 Let $\mathbf{bvec} : \mathcal{B}_n \rightarrow \mathbb{R}^r$ be defined as $\mathbf{bvec}(U) = [\sqrt{2}u_{1,1} \ \sqrt{4}u_{2,1} \ \cdots \ \sqrt{4}u_{n-1,1} \ \sqrt{2}u_{n,1} \ \sqrt{2}u_{2,2} \ \sqrt{4}u_{3,2} \ \cdots \ \sqrt{4}u_{n-1,2} \ \cdots \ \sqrt{g}u_{m,m}]^T$ for any $U \in \mathcal{B}_n$, where $m = n/2$ and $g = 2$ if n is even and $m = (n + 1)/2$ and $g = 4$ if n is odd.

It is clear that \mathbf{bvec} is a linear operator taking the set of all $n \times n$ real bisymmetric matrices to \mathbb{R}^r . The characterizations of \mathbf{bvec} are given by the following lemma:

Lemma 3.3 Given the operator \mathbf{bvec} , defined in the above definition, the following conditions hold: For any $W, P \in \mathcal{B}_n$

- (1) $W \bullet W = \mathbf{bvec}^T(W)\mathbf{bvec}(W)$.
- (2) $\|W - P\|_F^2 = \mathbf{bvec}^T(W - P)\mathbf{bvec}(W - P)$.

Proof It is clear from the definition of the operator \mathbf{bvec} that Part 1 is satisfied. Part 2 is a consequence of Part 1. □

From Part 1, it is clear that \mathbf{bvec} is an isometry. To take advantage of the above lemma, we need G to be bisymmetric. Projecting G onto \mathcal{B}_n using the above orthogonal projection given in (2.2) makes a bisymmetric matrix, say \bar{G} . In the following proposition, we show that the closest bisymmetric positive semi-definite matrix to \bar{G} is also the nearest to G .

Proposition 3.4 Given an orthogonal projection \bar{G} of G onto \mathcal{B}_n and let $B(b)$ be the closest bisymmetric positive semi-definite matrix to \bar{G} , then $B(b)$ is so for G .

Proof The proof is complete if \bar{G} is positive semi-definite. If not, then for any $T \in \mathcal{B}_n$, we have $(G - \bar{G})^T \bullet (\bar{G} - T) = 0$. This is true because \bar{G} is the orthogonal projection of G . Thus, $\|G - T\|_F^2 = \|G - \bar{G}\|_F^2 + \|\bar{G} - T\|_F^2$. \square

As a result of this proposition, the following problem is equivalent to (1.5):

$$\begin{aligned} & \text{minimize } \|\bar{G} - B(b)\|_F \\ & \text{subject to } B(b) \in \mathcal{B}_n, \quad B(b) \geq 0. \end{aligned} \tag{3.4}$$

From Lemma 3.1, the following are equivalences (for $t \geq 0 \in \mathbb{R}$):

$$\begin{aligned} & \|\bar{G} - B(b)\|_F^2 \leq t \\ \Leftrightarrow & \mathbf{bvec}^T(\bar{G} - B(b))\mathbf{bvec}(\bar{G} - B(b)) \leq t \text{ by Lemma 3.3} \\ \Leftrightarrow & t - \mathbf{bvec}^T(\bar{G} - B(b))\mathbf{I}\mathbf{bvec}(\bar{G} - B(b)) \leq 0 \\ \Leftrightarrow & \begin{bmatrix} \mathbf{I} & \mathbf{bvec}(\bar{G} - B(b)) \\ \mathbf{bvec}^T(\bar{G} - B(b)) & t \end{bmatrix} \geq 0 \text{ by Lemma 3.1.} \end{aligned}$$

Therefore, we have the following SDP problem:

$$\begin{aligned} (SDB) \quad & \text{minimize } t, \\ & \text{subject to } \begin{bmatrix} t & 0 & 0 \\ 0 & B(b) & 0 \\ 0 & 0 & \bar{V} \end{bmatrix} \geq 0, \end{aligned} \tag{3.5}$$

where

$$\bar{V} = \begin{bmatrix} \mathbf{I} & \mathbf{bvec}(\bar{G} - B(b)) \\ \mathbf{bvec}^T(\bar{G} - B(b)) & t \end{bmatrix}.$$

The dimensions of this problem are r and $n + r + 2$. These dimensions are much better than (3.3) where $r = mn - k$, $m = n/2$ and $k = n(n - 2)/4$ if n is even and $m = (n + 1)/2$ and $k = (n - 1)(n + 1)/4$ if n is odd.

3.3 Formulation III (SDQ)

An alternative method for formulating (1.5) is by means of the Frobenius norm definition:

$$\|G - B(b)\|_F^2 = y^T P y + 2q^T y + s,$$

where

$$\begin{aligned} y &= [b_1 \ b_2 \ \dots \ b_r]^T, \\ P &= \text{diag}([2 \ 4 \ \dots \ 4 \ 2 \ 2 \ 4 \ \dots \ 2]) \quad \text{and} \quad s = \|G\|_F^2 \\ q &= [-2G_{1,1}, -4G_{1,2} \ \dots \ -2G_{1,n}, \ -2G_{2,2}, \ -4G_{2,3} \ \dots \ -2G_{r,r}]. \end{aligned}$$

For a nonnegative real scalar t , we have the following problem:

$$\begin{aligned} & \|G - B\|_F^2 \leq t \\ \Leftrightarrow & y^T P y + 2q^T y + s \leq t \\ \Leftrightarrow & (P^{1/2}y)^T (P^{1/2}y) + 2q^T y + s \leq t \\ \Leftrightarrow & t - 2q^T y - s - (P^{1/2}y)^T \mathbf{I} (P^{1/2}y) \geq 0 \\ \Leftrightarrow & \begin{bmatrix} \mathbf{I} & (P^{1/2}y) \\ (P^{1/2}y)^T & t - 2q^T y - s \end{bmatrix} \geq 0. \end{aligned}$$

Therefore, we deduce the following SDP problem:

$$\begin{aligned}
 (SDQ) \quad & \text{minimize } t, \\
 & \text{subject to } \begin{bmatrix} t & 0 & 0 \\ 0 & B(b) & 0 \\ 0 & 0 & Q \end{bmatrix} \succeq 0,
 \end{aligned} \tag{3.6}$$

where

$$Q = \begin{bmatrix} I & (P^{1/2}y) \\ (P^{1/2}y)^T & t - 2q^T y - s \end{bmatrix}.$$

This is an SDP problem in the dual form (3.2) and the dimensions of this problem are $r + 1$ and $n + r + 2$. In spite of the fact that the dimensions of problem (3.5) is the same as the dimensions of problem (3.6), the latter is less efficient if we solve it using SDP method, especially when G is large in size. It has been found that in practice, as we will see later in Sect. 5, the performance of SDQ formula is not as efficient as in SDB formula. The reason for this inefficiency is the matrix P being of full rank which makes the system badly conditioned. We can develop a more efficient method for this formula by reformulating it over the second-order cone as described in Sect. 4, (see [12]).

Formula SDQ appears to be straightforward. Nonetheless, we found that using this SDQ formula to solve related problems was not an adequate option. In the next section, we will explain the cause for that when we speak about mixed SDP and second-order cone programming method. In Sect. 5, we will understand this fact about SDQ formula when we use the formula to solve large numerical examples with $n > 50$. The SDV formula does not contend favorably with the other two SDB and SDQ formulations due to the quantity of work per one iteration of interior-point method which solve SDV formula. It is $\mathcal{O}(n^6)$, where n in the dimension of G and $\mathcal{O}(\cdot)$ is the order of convergence. The SDV formulation is even slower than the projection method in some cases. Hence, using the SDV formulation to solve (1.5) is time consuming. The above discussion makes SDB formula as the best choice since we anticipate good performance while it does not have the illness of SDQ nor the large size of SDV.

4 Mixed semidefinite and second-order cone approach

Now, we explain the primal and the dual mixed semi-definite and second-order cone problem (SOCP) which is of the form:

$$\begin{aligned}
 (P') \quad & \text{minimize} && C_S \bullet X_S + C_Q^T X_Q \\
 & \text{subject to} && (D_S)_i \bullet X_S + (D_Q)_i^T X_Q = b_i, \quad i = 1, 2, \dots, m \\
 & && X_S \succeq 0, X_Q \succeq_Q 0.
 \end{aligned} \tag{4.1}$$

The variables are $X_S \in \mathcal{S}_n$ and $X_Q \in \mathbb{R}^k$. The given data matrices are $C_S, (D_S)_i \in \mathcal{S}_n$ and $C_Q, (D_Q)_i \in \mathbb{R}^k, \forall i$. In the two above inequalities each has a different explication : $X_S \succeq 0$ means $X_S \in \mathcal{S}_n^+$ and $X_Q \succeq_Q 0$ means that $X_Q \in \mathcal{Q}_k$.

The dual problem of (4.1) is:

$$\begin{aligned}
 (D') \quad & \text{maximize} && b^T y \\
 & \text{subject to} && \sum_{i=1}^m y_i (D_S)_i \leq C_S \\
 & && \sum_{i=1}^m y_i (D_Q)_i \leq_Q C_Q.
 \end{aligned} \tag{4.2}$$

Here, $y \in \mathbb{R}^m$ is the variable.

The objective function of problem (1.5) can now be rewritten as a dual SOCP in three different ways.

4.1 Formulation IV (SQV)

One direction is to define $\|G - B(b)\|_F$ as

$$\|G - B(b)\|_F = \|\mathbf{vec}(G - B(b))\|_2.$$

Hence, if we place $\|G - B(b)\|_F \leq t$ for t belong to \mathbb{R}^+ , the second-order cone definition gives us

$$\begin{bmatrix} t \\ \mathbf{vec}(G - B(b)) \end{bmatrix} \in \mathcal{Q}_{1+n^2}.$$

Therefore, we have the following equivalent problem to (1.5):

$$\begin{aligned} (SQV) \quad & \text{minimize } t, \\ & \text{subject to } \begin{bmatrix} t & 0 \\ 0 & B(b) \end{bmatrix} \succeq 0 \begin{bmatrix} t \\ \mathbf{vec}(G - B(b)) \end{bmatrix} \succeq_{\mathcal{Q}} 0, \end{aligned} \tag{4.3}$$

where $t \in \mathbb{R}^+$. The above problem is in the shape of problem (4.2). The second-order cone constraint is the difference between this form and SDV. The dimensions of this SQV problem are $r + 1$, the SDP part is $n + 1$ and the SOCP part is $n^2 + 1$. This makes us anticipate much less efficiency from SQV when we execute it.

4.2 Formulation V (SQQ)

The second formula is as established in Sect. 3.3, i.e.

$$\|G - B(b)\|_F^2 = y^T P y + 2q^T y + s.$$

Therefore, we have the following problem which is equivalent to problem (1.5)

$$\begin{aligned} & \text{minimize } y^T P y + 2q^T y + s \\ & \text{subject to } B(b) \in \mathcal{B}_n, \quad B(b) \succeq 0. \end{aligned} \tag{4.4}$$

But

$$y^T P y + 2q^T y + s = \|P^{1/2} y + P^{-1/2} q\|_2^2 + s - q^T P^{-1} q.$$

Now, we optimize $\|G - B(b)\|_F^2$ by minimizing $\|P^{1/2} y + P^{-1/2} q\|_2$. Then, we have the following equivalent problem:

$$\begin{aligned} (SQQ) \quad & \text{minimize } t, \\ & \text{subject to } \begin{bmatrix} t & 0 \\ 0 & B(b) \end{bmatrix} \succeq 0 \begin{bmatrix} t \\ P^{1/2} y + P^{-1/2} q \end{bmatrix} \succeq_{\mathcal{Q}} 0, \end{aligned} \tag{4.5}$$

where $t \in \mathbb{R}^+$. The above problem is in the shape of problem (4.2). The second-order cone constraint is the difference between this form and SDQ. The dimensions of this SQQ problem are $r + 1$, the SDP part is $n + 1$ and the SOCP part is $r + 1$. It may be noticed that we did not speak about the constraint of $B(b)$ being bisymmetric. This is because the structure of the bisymmetric matrix $B(b)$ is embedded inside the other constraints.

Table 1 $\|G - B^*\|_F$ among all methods

n	20	40	50	70	90	110	130	150
	0.2999	0.6054	0.7845	1.0787	1.4002	1.71605	1.97994	2.30513

4.3 Formulation VI (SQB)

The last formula will take advantage and feature the bisymmetric structure of $B(b)$ explicitly. In Sect. 3, we introduced the operator vectorization \mathbf{bvec} on bisymmetric matrices. This operator will be used to develop SQB formula. Lemma 3.3 gives us the following:

$$\|\bar{G} - B(b)\|_F = \|\mathbf{bvec}(\bar{G} - B(b))\|_2$$

where $\bar{G} = P_B(G)$, which leads to:

$$(SQB) \quad \begin{aligned} & \text{minimize } t, \\ & \text{subject to } \begin{bmatrix} t & 0 \\ 0 & B(b) \end{bmatrix} \succeq 0 \left[\mathbf{bvec} \begin{bmatrix} t \\ \bar{G} - B(b) \end{bmatrix} \right] \succeq_Q 0. \end{aligned} \quad (4.6)$$

The second-order cone dimension in this form is $r + 1$, which is the same as that of SQQ.

The mixed formulations are expected to be more efficient in practice than SDP-only formulas, particularly the SQQ and SQB which have the least dimension in the second-order cone constraint. For interior point method, the SOCP has superior worst-case complexity more than the SDP method. Nevertheless, SDB has a much less SDP dimension with no weakness like SDQ has, and that causes SDB a preferable choice among other SDP. This is due to the economical vectorization operator \mathbf{bvec} . It was clear from practical experiments that the SQB formula shows competitive behaviour over SQQ and similar to SDB (see Sect. 5).

5 Numerical results

In this section, we compare and present the performance of the methods we discussed in previous sections. First, we present numerical results for the projection method and then we use NT-direction of the interior-point primal-dual path-following method. Next, we present numerical results for all the six different formulas of Sects. 3 and 4, and compare them with the alternating projection method.

To implement the modified alternating projection method, a Matlab code was written and the iteration is stopped when $\|P_B(P_S(G_j)) - P_S(G_j)\|_F \leq 10^{-5}$. For the six SDP and SOCP formulas, we used the software SDPT3 ver. 3.0 [17] because of its stability and its ability to utilize sparsity very efficiently.

Problem (1.5) was transformed into six different formulas as explained in Sects. 3 and 4. For each formula, we wrote a Matlab code. This Matlab code transforms the problem and passes it through to SDPT3 for a first run. A second run is done with the minimal iterate from the first run being the starting initial point. We repeat this process until no more progress is detected. We execute all the numerical experiment in this section using Matlab 9.0.

We applied all approaches to problems starting from small dense problems with $n = 10$ up to a large problems with $n = 150$. We apply the methods to obtain results as follows: we construct a matrix B positive definite bisymmetric randomly, then we perturb this matrix by adding random noise matrix N to B , where elements of B changes between -10.0 and 10.0 . Then, the problem is to retrieve the matrix before the matrix noise was added. The optimal solution is found for all the cases with high accuracy, at least seven decimals, except for the projection method where we stop with five decimals of accuracy. Table 1 shows how close, in Frobenius norm for selected test problem and the minimal solution of each method, B^* to the data matrix G . We show in Fig. 1 the comparison results, comparing the CPU time in \ln of seconds against the size of the data matrix G . The correlation between the size of the matrix and the CPU time is shown in Fig. 1.

The SDV formulation does not compete favorably with the other formulations due to the volume of work at each iteration of interior-point methods which solve SDV formula in $\mathcal{O}(n^6)$, where $\mathcal{O}(\cdot)$ is the order of convergence. The SDV formulation is even sometimes slower than the projection method. Hence, using the SDV formulation to solve (1.5) is time consuming. This leaves us with SDB and SQB formula in which we



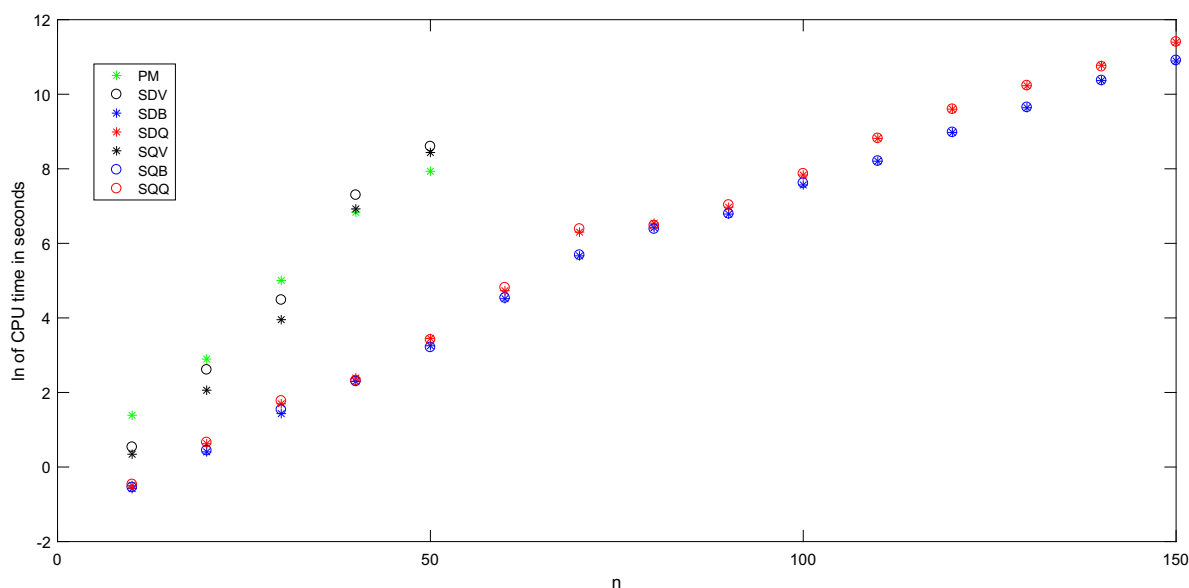


Fig. 1 Comparing CPU time in minutes for all seven approaches

anticipate good performance since it does not have the weakness of SDQ or SQQ nor the huge size of SDV or SQV. However, we practically found that all formulations work well except for the SDV and SQV.

Acknowledgements The author is grateful to King Fahd University of Petroleum and Minerals for providing excellent research facilities.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

1. Aitken, A.C.: Note on a special persymmetric determinant. *Ann. Math.* **2**(32), 461–462 (1931)
2. Al-Homidan, S.: Low rank methods for solving the nearest correlation matrix problem. *J. Nonlinear Convex Anal.* **19**(6), 881–892 (2018)
3. Al-Homidan, S.: Structure method for solving the nearest Euclidean distance matrix problem. *J. Inequal. Appl.* **1**, 491 (2014)
4. Al-Homidan, S.: Approximate Toeplitz problem using semidefinite programming. *J. Optim. Theory Appl.* **135**(3), 583–598 (2007)
5. Alizadeh, F.; Haerberly, J.; Overton, M.: Primal-dual interior-point methods for semi-definite programming: convergence rates, stability and numerical results. *SIAM J. Optim.* **8**, 746–768 (1998)
6. Cantoni, A.; Butler, P.: Properties of the eigenvectors of persymmetric matrices with applications to communication theory. *IEEE Trans. Commun.* **COM 24**(8), 804–809 (1976)
7. Devillet, J.: Bisymmetric and quasitrivial operations: characterizations and enumerations. *Aequat. Math.* **93**(3), 501–526 (2019)
8. Dykstra, R.L.: An algorithm for restricted least squares regression. *J. Am. Stat.* **78**, 839–842 (1983)
9. Hajarian, M.; Chronopoulos, A.: Least-squares partially bisymmetric solutions of coupled Sylvester matrix equations accompanied by a prescribed submatrix constraint. *Math. Methods Appl. Sci.* **44**(6), 4297–4315 (2021)
10. Higham, N.: Computing a nearest symmetric positive semi-definite matrix. *Linear Algebra Appl.* **103**, 103–118 (1988)



11. Liao, A.P.; Xie, D.: Least-squares solution for a class of inverse eigenvalue problems for bisymmetric nonnegative definite matrices. *Math. Numer. Sin.* **23**, 209–218 (2001)
12. Nesterov, Y.; Nemirovskii, A.: *Interior Point Polynomial Methods in Convex Programming*. SIAM, Philadelphia (1994)
13. Pressman, I.S.: Matrices with multiple symmetry properties: applications of centrohermitian and perhermitian matrices. *Linear Algebra Appl.* **284**, 239–258 (1998)
14. Sablik, M.: Bisymmetric functionals revisited or a converse of the Fubini theorem. *Aequat. Math.* **94**(4), 629–649 (2020)
15. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **14**, 625–653 (1999)
16. Todd, M.: Semidefinite optimization. *Acta Numer.* **10**, 515–560 (2001)
17. Tütüncü, R.; Toh, K.; Todd, M.: Solving semi-definite-quadratic-linear programs using SDPT3. *Math. Program.* **95**, 189–217 (2003)
18. von Neuman, J. *Functional Operators II, The geometry of orthogonal spaces*. Annals of Math. studies No.22, Princeton University Press, Princeton (1950)
19. Wang, D.; Li, Y.; Ding, W.: The least squares bisymmetric solution of quaternion matrix equation $AXB = C$. *AIMS Math.* **6**(12), 13247–13257 (2021)
20. Xie, D.; Sheng, Y.; Zhang, Z.: Computing the nearest bisymmetric positive semi-definite matrix under the spectral restriction. *Numer. Math. J. Chin. Univ.* **12**, 71–82 (2003)
21. Yan, T.; Ma, C.: An iterative algorithm for solving a class of generalized coupled Sylvester-transpose matrix equations over bisymmetric or skew-anti-symmetric matrices. *J. Appl. Anal. Comput.* **10**(4), 1282–1310 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.