ROW BY ROW METHODS FOR SEMIDEFINITE PROGRAMMING

ZAIWEN WEN[†], DONALD GOLDFARB[†], SHIQIAN MA[†], AND KATYA SCHEINBERG

April 28, 2009

Abstract. We present a row-by-row (RBR) method for solving semidefinite programming (SDP) problem based on solving a sequence of problems obtained by restricting the n-dimensional positive semidefinite constraint on the matrix X. By fixing any $(n-1)$-dimensional principal submatrix of X and using its (generalized) Schur complement, the positive semidefinite constraint is reduced to a simple second-order cone constraint. When the RBR method is applied to solve the maxcut SDP relaxation, the optimal solution of the RBR subproblem only involves a single matrix-vector product which leads to a simple and very efficient method. To handle linear constraints in generic SDP problems, we use an augmented Lagrangian approach. Specialized versions are presented for the maxcut SDP relaxation and the minimum nuclear norm matrix completion problem since closed-form solutions for the RBR subproblems are available. Finally, numerical results on the maxcut SDP relaxation and matrix completion problems are presented to demonstrate the robustness and efficiency of our algorithm.

Key words. semidefinite programming, row by row method, (block) coordinate descent method, nonlinear Gauss-Seidel method, alternating direction method, matrix completion, maximum cut

AMS subject classifications. 90C06, 90C22, 90C30, 90C35

1. Introduction.

In this paper we present a new method for solving semidefinite programming (SDP) problems. These convex optimization problems are solvable in polynomial time by interior point methods [25, 27, 28]. Unfortunately, however, in practice large scale SDPs are quite difficult to solve because of the very large amount of work required by each iteration of an interior point method. Most of these methods form a positive definite $m \times m$ matrix M, where m is the number of constraints in the SDP, and then compute the search direction by finding the Cholesky factorization of M. Since m can be $O(n^2)$ when the unknown positive semidefinite matrix is $n \times n$, it can take $O(n^6)$ arithmetic operations to do this. Consequently, this becomes impractical both in terms of the time and the amount of memory required ($O(m^2)$) when n is much larger than one hundred and m is much larger than a few thousand. Moreover forming M itself can be prohibitively expensive unless m is not too large or the constraints in the SDP are very sparse [11]. Although the computational complexity of the new method presented here is not polynomial, each of its iterations can be executed much more cheaply than in an interior point algorithm. This enables it to solve very large SDPs efficiently. Preliminary numerical testing verifies this. For example, variants of our new method produce highly accurate solutions to maxcut SDP relaxation problems involving matrices of size 4000×4000 in less than 5.25 minutes and nuclear norm matrix completion SDPs involving matrices of size 1000×1000 in less than 1 minute on a 3.4 GHZ workstation. If only moderately accurate solutions are required (i.e., a relative accuracy of the order of 10^{-3}) then less than 45 and 10 seconds, respectively, is needed.

Our method is based upon the well known relationship between the positive semidefiniteness of a symmetric matrix and properties of the Schur complement of a sub-matrix of that matrix [29]. We note that Schur complements play an important role in semidefinite programming and related optimization problems. For example, they are often used to formulate problems as SDPs [7, 27]. In [1, 13, 14] they are used to reformulate certain SDP problems as second-order cone programs (SOCPs). More recently, they have been used by Banerjee, El Ghaoui and d'Aspremont [2] to develop a method for solving a maximum likelihood

[†]Department of Industrial Engineering and Operations Research, Columbia University, New York, 10027, U.S.A. (zw2109@columbia.edu, goldfarb@columbia.edu, sm2756@columbia.edu). Research supported in part by NSF Grant DMS 06-06712, ONR Grant N00014-08-1-1118 and DOE Grant DE-FG02-08ER58562.

estimation problem whose objective function involves the log determinant of a positive semidefinite matrix. That method is closely related to the method proposed here.

As in the method proposed in Banerjee et. al [2], we use Schur complements to develop an *overlapping* block coordinate descent method. The coordinates (i.e., variables) in each iteration of our method correspond to the components of a single row (column) of the unknown semidefinite matrix. Since every row (column) of a symmetric matrix contains one component of each of the other rows (columns), the blocks in our method overlap. The convergence properties of block coordinate descent methods [6, 15, 16, 19, 26], which are also referred to as nonlinear Gauss-Seidel methods and subspace correction methods [24], have been intensively studied. In the case of (single) coordinate descent, convergence is not guaranteed for nonconvex functions [22]. Convergence can be proved under fairly mild conditions when the objective function is convex. As we shall see below, the convergence result in [6] can be extended to the case of overlapping blocks. However, they do not apply to the case where constraints couple the variables between different blocks.

1.1. Preliminaries. The set of $n \times n$ symmetric matrices is denoted by S^n and the set of $n \times n$ symmetric positive semidefinite (positive definite) matrices is denoted by S^n_+ (S^n_{++}). The notation $X \succeq 0$ ($X \succ 0$) means that X is positive semidefinite (positive definite).

Our row-by-row (**RBR**) method is motivated by the following fact about the Schur complement of a positive definite matrix. Assume the matrix $X \in S^n$ can be partitioned as

$$X := \begin{pmatrix} \xi & y^\top \\ y & B \end{pmatrix}, \tag{1.1}$$

where $\xi \in \mathbb{R}$, $y \in \mathbb{R}^{n-1}$ and $B \in S^{n-1}$ is nonsingular. Since X can be factorized as

$$X = \begin{pmatrix} 1 & y^\top B^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \xi - y^\top B^{-1} y & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} 1 & 0 \\ B^{-1} y & I \end{pmatrix}, \tag{1.2}$$

the positive definite constraint $X \succ 0$ is equivalent to

$$X \succ 0 \Longleftrightarrow B \succ 0 \text{ and } (X/B) := \xi - y^\top B^{-1} y > 0, \tag{1.3}$$

where (X/B) is called the Schur complement of X with respect to B. If the matrix B is singular, the generalized Schur complement of B in X is defined as

$$(X/B) := \xi - y^\top B^\dagger y, \tag{1.4}$$

where $B^\dagger$ is the Moore-Penrose pseudo-inverse of B and the following theorem generalizes (1.3).

THEOREM 1.1. *([29], Theorems 1.12 and 1.20) Let $X \in S^n$ be a symmetric matrix partitioned as (1.1) in which ξ is a scalar and $B \in S^{n-1}$. The generalized Schur complement of B in X is defined as $(X/B) := \xi - y^\top B^\dagger y$, where $B^\dagger$ is the pseudo-inverse of B. Then the following holds.*
1) If B is nonsingular, then $X \succ 0$ if and only if both $B \succ 0$ and $(X/B) > 0$.
2) If B is nonsingular, then $X \succeq 0$ if and only if both $B \succ 0$ and $(X/B) \geq 0$.
3) $X \succeq 0$ if and only if $B \succeq 0$, $(X/B) \geq 0$ and $y \in \mathcal{R}(B)$, where $\mathcal{R}(B)$ is the range space of B.

We adopt the following notation in this paper. Given a matrix $A \in \mathbb{R}^{n\times n}$, we denote the (i,j)-th entry of A by either $A_{i,j}$ or $A(i,j)$. Let α and β be given index sets, i.e., subsets of $\{1, 2, \cdots, n\}$. We denote the cardinality of α by $|\alpha|$ and its complement by $\alpha^c := \{1, 2, \cdots, n\}\backslash\alpha$. Let $A_{\alpha,\beta}$ denote the submatrix of A

with rows indexed by α and columns indexed by β, i.e.,

$$A_{\alpha,\beta} := \begin{pmatrix} A_{\alpha_1,\,\beta_1} & \cdots & A_{\alpha_1,\,\beta_{|\beta|}} \\ \vdots & & \vdots \\ A_{\alpha_{|\alpha|},\,\beta_1} & \cdots & A_{\alpha_{|\alpha|},\,\beta_{|\beta|}} \end{pmatrix}.$$

Without introducing any confusion, we write i for the index set $\{i\}$ and denote the complement of $\{i\}$ by $i^c := \{1, 2, \cdots, n\}\backslash\{i\}$. Hence, A_{i^c,i^c} is the submatrix of A that remains after removing its i-th row and column, and $A_{i^c,i}$ is the ith column of the matrix A without the element $A_{i,i}$. The inner product between two matrices C and X is defined as $\langle C, X\rangle := \sum_{jk} C_{j,k}X_{j,k}$ and the trace of X is defined as $Tr(X) = \sum_{i=1}^{n} X_{ii}$. The vector $\begin{pmatrix} x \\ y \end{pmatrix}$ obtained by stacking the vector $x \in \mathbb{R}^p$ on the top of the vector $y \in \mathbb{R}^q$ is also denoted by $[x; y] \in \mathbb{R}^{p+q}$.

The rest of this paper is organized as follows. In section 2, we present a prototype of the RBR method for solving a general SDP. In section 3, the RBR method is specialized for solving SDPs with only diagonal element constraints. We interpret this RBR method in terms of the logarithmic barrier function in section 3.1 and prove convergence to a global minimizer. To handle general linear constraints, we apply the RBR method in section 4 to a sequence of unconstrained problems using an augmented Lagrangian function approach. Specialized versions for the maxcut SDP relaxation and the minimum nuclear norm matrix completion problem are presented in sections 4.2 and 4.3, respectively Finally, numerical results for the maxcut and matrix completion problems, are presented in section 5 to demonstrate the robustness and efficiency of our algorithms.

2. A row-by-row method prototype.

Consider the semidefinite programming (SDP) problem

$$\begin{aligned} \min_{X\in S^n} \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \mathcal{A}(X) = b, \quad X \succeq 0, \end{aligned} \tag{2.1}$$

where the linear map $\mathcal{A}(\cdot) : S^n \to \mathbb{R}^m$ and its adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to S^n$ are defined by

$$\mathcal{A}(X) := \begin{pmatrix} \langle A^{(1)}, X\rangle \\ \cdots \\ \langle A^{(m)}, X\rangle \end{pmatrix}, \text{ and } \mathcal{A}^*(y) := \sum_{i=1}^{m} y_i A^{(i)}, \tag{2.2}$$

the matrices $C, A^{(i)} \in S^n$, and the vector $b \equiv (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$ are given, and the unknown matrix $X \in S^n_+$. Throughout this paper, the following Slater condition for (2.1) is assumed to hold.

Assumption 2.1. *Problem* (2.1) *satisfies the Slater condition:*

$$\begin{cases} \mathcal{A} : S^n \to \mathbb{R}^m \text{ is onto }, \\ \exists X^1 \in S^n_{++} \text{ such that } \mathcal{A}(X^1) = b. \end{cases} \tag{2.3}$$

Given a strictly feasible solution $X^k \succ 0$, we can construct a second-order cone programming (SOCP) restriction for the SDP problem (2.1) as follows. Fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{1^c,1^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{1,1}$ and $X_{1^c,1}$ (i.e., row 1/column 1), respectively, that is $X := \begin{pmatrix} \xi & y^\top \\ y & B \end{pmatrix} := \begin{pmatrix} \xi & y^\top \\ y & X^k_{1^c,1^c} \end{pmatrix}$. It then follows from Theorem 1.1 that $X \succeq 0$ is equivalent to $\xi - y^\top B^{-1} y \geq 0$, and hence, the SDP problem (2.1) becomes

$$\begin{aligned} \min_{[\xi;y]\in\mathbb{R}^n} \quad & \widetilde{c}^\top [\xi; y] \\ \text{s.t.} \quad & \widetilde{A}\,[\xi; y] = \widetilde{b}, \\ & \xi - y^\top B^{-1} y \geq 0, \end{aligned} \tag{2.4}$$

where $\nu > 0$, $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ are defined as follows using the subscript $i = 1$:

$$\widetilde{c} := \begin{pmatrix} C_{i,i} \\ 2C_{i^c,i} \end{pmatrix}, \quad \widetilde{A} := \begin{pmatrix} A^{(1)}_{i,i} & 2A^{(1)}_{i,i^c} \\ \cdots & \cdots \\ A^{(m)}_{i,i} & 2A^{(m)}_{i,i^c} \end{pmatrix} \text{ and } \widetilde{b} := \begin{pmatrix} b_1 - \left\langle A^{(1)}_{i^c,i^c}, B \right\rangle \\ \cdots \\ b_m - \left\langle A^{(m)}_{i^c,i^c}, B \right\rangle \end{pmatrix}. \tag{2.5}$$

If we let $LL^\top = B$ be the Cholesky factorization of B and introduce a new variable $z = L^{-1}y$, the Schur complement constraint $\xi - y^\top B^{-1} y \geq 0$ is equivalent to the linear constraint $Lz = y$ and the rotated second-order cone constraint $\|z\|_2^2 \leq \xi$. Furthermore, positive definiteness of the solution X can be maintained if we replace the 0 on the right hand side of the Schur complement constraint in subproblem (2.4) by $\nu > 0$, i.e., if we replace subproblem (2.4) by

$$\begin{aligned} \min_{[\xi;y]\in\mathbb{R}^n} \quad & \widetilde{c}^\top [\xi; y] \\ \text{s.t.} \quad & \widetilde{A}\,[\xi; y] = \widetilde{b}, \\ & \xi - y^\top B^{-1} y \geq \nu, \end{aligned} \tag{2.6}$$

Clearly, similar problems can be constructed if for any i, $i = 1, \cdots, n$, all elements of X^k other than those in the i-th row/column are fixed and only the elements in the i-th row/column are treated as unknowns.

Remark 2.2. *For any $i \in \{1, \cdots, n\}$, the Schur complement of X_{i^c,i^c} in X is $(X/X_{i^c,i^c}) := X_{i,i} - X^\top_{i^c,i} X^\dagger_{i^c,i^c} X_{i^c,i}$. There exists a permutation matrix P of the rows and columns of X that put X_{i^c,i^c} into the lower right corner of X, leaving the rows and columns of X_{i,i^c} and $X_{i^c,i}$ in the same increasing order in X, that is $P^\top X P = \begin{pmatrix} X_{i,i} & X^\top_{i^c,i} \\ X_{i^c,i} & X_{i^c,i^c} \end{pmatrix}$. Therefore, the Schur complement of X_{i^c,i^c} in X is $((P^\top X P)/X_{i^c,i^c})$.*

We now present our row-by-row (**RBR**) method for solving (2.1). Starting from a positive definite feasible solution X^1, we update one row/column of the solution X at each of n inner steps by solving subproblems of the form (2.6). This procedure from the first row to the n-th row is called a *cycle*. At the first step of the k-th cycle, we fix $B := X^k_{1^c,1^c}$, and solve subproblem (2.6), whose solution is denoted by $[\xi; y]$. Then the first row/column of X^k is replaced by $X^k_{1,1} := \xi$ and $X^k_{1^c,1} := y$. Similarly, we set $B := X^k_{i^c,i^c}$ in the i-th inner iteration and assign the parameters $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ according to (2.5). Then the solution $[\xi; y]$ of (2.6) is used to set $X^k_{i,i} := \xi$ and $X^k_{i^c,i} := y$. The k-th cycle is finished after the n-th row/column is updated. Then we set $X^{k+1} := X^k$ and repeat this procedure until the relative decrease in the objective function on a cycle becomes smaller than some tolerance ϵ. Our RBR method prototype is outlined in Algorithm 1.

We also denote by $X^{k,j}$ the solution X^k at the end of the j-th inner iteration in the k-th cycle and use the convention that $X^{k,0} := X^k$ and $X^{k+1} := X^{k,n}$. Our RBR method is similar to a block *Gauss-Seidel* method for solving a system of linear equations and a block coordinate descent method (sometimes referred

Algorithm 1: A row-by-row (RBR) method prototype

```
Set $X^1 \succ 0$, $\nu \geq 0$, $k := 1$ and $\epsilon \geq 0$. Compute $F^0 := \langle C, X^1 \rangle$ and set $F^1 := +\infty$.
while $\frac{F^{k-1} - F^k}{\max\{|F^{k-1}|, 1\}} \geq \epsilon$ do
    for $i = 1, \cdots, n$ do
        Set $B := X^k_{i^c, i^c}$ and the parameters $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ according to (2.5).
        Solve the SOCP subproblem (2.6) whose solution is denoted by $\xi$ and $y$.
        Update $X^k_{i,i} := \xi$, $X^k_{i^c,i} := y$ and $X^k_{i,i^c} := y^\top$.
    Compute $F^k := \langle C, X^k \rangle$. Set $X^{k+1} := X^k$ and $k := k + 1$.
```

to as a nonlinear Gauss-Seidel method) for nonlinear programming, except that because of the symmetry of X, the blocks in our method overlap. Specifically, exactly one of the variables in any two inner iterations of the RBR method overlap. For example, $X_{1,2}$ is a variable in the three dimensional problem in both the first and second inner iterations:

$$X^{k,1} := \begin{pmatrix} X^{k,1}_{1,1} & X^{k,1}_{1,2} & X^{k,1}_{1,3} \\ X^{k,1}_{1,2} & X^{k,0}_{2,2} & X^{k,0}_{2,3} \\ X^{k,1}_{1,3} & X^{k,0}_{2,3} & X^{k,0}_{3,3} \end{pmatrix}, \quad X^{k,2} := \begin{pmatrix} X^{k,1}_{1,1} & \boxed{X^{k,2}_{1,2}} & X^{k,1}_{1,3} \\ \boxed{X^{k,2}_{1,2}} & X^{k,2}_{2,2} & X^{k,2}_{2,3} \\ X^{k,1}_{1,3} & X^{k,2}_{2,3} & X^{k,0}_{3,3} \end{pmatrix}.$$

3. The RBR method for SDP with only diagonal element constraints.

In this section, we consider an SDP whose constraints $\mathcal{A}(X) = b$ are of the form $X_{i,i} = b_i$, where $b_i > 0$, $i = 1, \cdots, n$. Without loss of generality, we assume that $b_i = 1$, for $i = 1, \cdots, n$, and study the problem

$$\begin{aligned} \min_{X \in S^n} \quad & \langle C, X \rangle \\ \text{s.t.} \quad & X_{ii} = 1, \quad i = 1, \cdots, n, \\ & X \succeq 0. \end{aligned} \tag{3.1}$$

Note that (3.1) is the well known SDP relaxation [18, 12, 8, 17, 3] for the maxcut problem, which seeks to partition the vertices of a graph into two sets so that the sum of the weighted edges connecting vertices in one set with vertices in the other set is maximized.

We now present the RBR subproblem for solving (3.1). Throughout the algorithm the diagonal elements of X are kept fixed at 1. At the ith step of the k-th cycle, we fix $B = X^k_{i^c,i^c}$, where X^k is the iterate at the $(i-1)$-st step of the k-th cycle. Here, we do not assume that B is positive definite and use the generalized Schur complement to construct the second-order cone constraint. Hence, the subproblem (2.6) for generic SDP is reduced to

$$\begin{aligned} \min_{y \in \mathbb{R}^{n-1}} \quad & \widehat{c}^\top y \\ \text{s.t.} \quad & 1 - y^\top B^\dagger y \geq \nu, \quad y \in \mathcal{R}(B), \end{aligned} \tag{3.2}$$

where $\widehat{c} := 2C_{i^c,i}$. Fortuitously, the optimal solution of (3.2) is determined by a single matrix-vector product as we can see from the following lemma.

Lemma 3.1. *If $\gamma := \widehat{c}^\top B \widehat{c} > 0$, the solution of problem* (3.2) *is given by*

$$y = -\sqrt{\frac{1-\nu}{\gamma}} B\widehat{c}. \tag{3.3}$$

Otherwise, the solution is $y = 0$.

Proof. Since the matrix $B \in S^n_+$ is real symmetric and the rank of B is $r > 0$, B has the spectral decomposition

$$B = Q\Lambda Q^\top = \begin{pmatrix} Q_r & Q_l \end{pmatrix} \begin{pmatrix} \Lambda_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_r^\top \\ Q_l^\top \end{pmatrix} = Q_r \Lambda_r Q_r^\top. \tag{3.4}$$

where Q is an orthogonal matrix, $\Lambda = \operatorname{diag}(\lambda_1, \cdots, \lambda_r, 0, \cdots 0)$, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$.

Hence, the Moore-Penrose pseudo-inverse of B is

$$B^\dagger = \begin{pmatrix} Q_r & Q_l \end{pmatrix} \begin{pmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Q_r^\top \\ Q_l^\top \end{pmatrix} = Q_r \Lambda_r^{-1} Q_r^\top.$$

Let $z = Q^\top y =: [z_r; z_l]$. Since $y \in \mathcal{R}(B)$ and $\mathcal{R}(B) = \mathcal{R}(Q_r)$, $z_l = 0$; hence, problem (3.2) is equivalent to

$$\begin{aligned} \min_{z_r \in \mathbb{R}^r} \quad & (Q_r^\top \widehat{c})^\top z_r \\ \text{s.t.} \quad & 1 - z_r^\top \Lambda_r^{-1} z_r \geq \nu, \end{aligned} \tag{3.5}$$

whose Lagrangian function is $\ell(z_r, \lambda) = (Q_r^\top \widehat{c})^\top z_r - \frac{\lambda}{2}(1 - \nu - z_r^\top \Lambda^{-1} z_r)$, where $\lambda \geq 0$. At an optimal solution z_r^* to (3.5),

$$\nabla_{z_r} \ell(z_r^*, \lambda^*) = Q_r^\top \widehat{c} + \lambda^* \Lambda_r^{-1} z_r^* = 0,$$

which implies $z_r^* = -\Lambda_r Q_r^\top \bar{c}/\lambda^*$. Since z_r^* is on the boundary of the constraint, i.e., $1 - (z_r^*)^\top \Lambda^{-1} z_r^* = \nu$, we obtain

$$1 - \frac{\widehat{c}^\top Q_r \Lambda_r \Lambda_r^{-1} \Lambda_r Q_r^\top \widehat{c}}{(\lambda^*)^2} = 1 - \frac{\gamma}{(\lambda^*)^2} = \nu.$$

Hence, if $\gamma > 0$, we obtain $\lambda^* = \sqrt{\gamma/(1-\nu)}$ and

$$y^* = Q_r z_r^* = -\sqrt{\frac{1-\nu}{\gamma}} Q_r \Lambda_r Q_r^\top \widehat{c} = -\sqrt{\frac{1-\nu}{\gamma}} B\widehat{c}.$$

Otherwise, $\lambda^* = 0$ and $y^* = 0$. □

We present the RBR method for (3.1) in Algorithm 2.

Algorithm 2 is extremely simple since only a single matrix-vector product is involved at each inner step. Numerical experiments show Algorithm 2 works fine if the initial solution X is taken as the identity matrix even if we take $\nu = 0$. However, the following is a 3×3 example which shows that if started at a rank one point that is not optimal, the RBR method using $\nu = 0$ either does not move away from the initial solution

Algorithm 2: A RBR method (PURE-RBR-M) for problem (3.1)

```
Set $X^1 \succ 0$, $\nu > 0$, $k := 1$ and $\epsilon \geq 0$. Compute $F^0 := \langle C, X^1 \rangle$ and set $F^1 := +\infty$.
while $\frac{F^{k-1} - F^k}{\max\{|F^{k-1}|, 1\}} \geq \epsilon$ do
    for $i = 1, \cdots, n$ do
        Compute $x := X^k_{i^c, i^c} C_{i^c, i}$ and $\gamma := x^\top C_{i^c, i}$. Set $X^k_{i,i} := 1$.
        if $\gamma > 0$ then compute $X^k_{i^c, i} := -\sqrt{\frac{1-\nu}{\gamma}} x$, else set $X^k_{i^c, i} := \mathbf{0}$.
    Compute $F^k := \langle C, X^k \rangle$. Set $X^{k+1} := X^k$ and $k := k + 1$.
```

Table 3.1

feasible rank-one and optimal solutions of (3.6)

solution	(x, y, z)	$\frac{1}{2}\langle C, X \rangle = \frac{3}{4}x - y - z$
$X^{(1)}$	(1,1,1)	-5/4
$X^{(2)}$	(-1,1,-1)	-3/4
$X^{(3)}$	(-1,-1,1)	-3/4
$X^{(4)}$	(1,-1,-1)	11/4
$X^{(5)}$	(-1/9,2/3,2/3)	-17/12

or it moves to a non-optimal rank-one solution and stays there. Let

$$X = \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \text{ and } C = \begin{pmatrix} 0 & 3/4 & -1 \\ 3/4 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix}. \tag{3.6}$$

The rank-one feasible solutions $X^{(1)}$, $X^{(2)}$, $X^{(3)}$ and $X^{(4)}$ and the rank-two optimal solution $X^{(5)}$ for this example are given in Table 3.1. Starting at $X^{(1)}$, each row-by-row minimization step leaves the matrix unchanged. Starting at $X^{(2)}$, $X^{(3)}$ or $X^{(4)}$, the row-by-row method moves to the point $X^{(1)}$ and then remains there. Interestingly, Algorithm 2 is able to find the optimal solution if the off-diagonal elements x, y and z of the rank-one solutions in Table 3.1 are scaled by 0.999 (or even by a number much closer to 1).

3.1. Interpretation of Algorithm 2 in terms of the logarithmic barrier function. In this subsection, we interpret Algorithm 2 as a variant of the row-by-row method applied to a logarithmic barrier function approximation to (3.1).

Lemma 3.2 below relates that the optimal solution (3.3) of the RBR subproblem (3.2) to the solution of the following logarithmic barrier function minimization

$$\min_{y \in \mathbb{R}^{n-1}} \quad \widehat{c}^\top y - \sigma \log(1 - y^\top B^\dagger y), \qquad \text{s.t. } y \in \mathcal{R}(B). \tag{3.7}$$

Lemma 3.2. *If $\gamma := \widehat{c}^\top B \widehat{c} > 0$, the solution of problem* (3.7) *is*

$$y = -\frac{\sqrt{\sigma^2 + \gamma} - \sigma}{\gamma} B\widehat{c}. \tag{3.8}$$

Hence, the subproblem (3.2) *has the same solution as* (3.7) *if $\nu = 2\sigma \frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$.*

Proof. Similar to Lemma 3.1, we have the spectral decomposition (3.4) of B. Let $z = Q^\top y =: [z_r; z_l]$. Since $y \in \mathcal{R}(B)$ and $\mathcal{R}(B) = \mathcal{R}(Q_r)$, we obtain $z_l = 0$ and hence $y = Q_r z_r$. Therefore, problem (3.7) is equivalent to

$$\min_{z_r} \quad (Q_r^\top \widehat{c})^\top z_r - \sigma \log(1 - z_r^\top \Lambda_r^{-1} z_r), \tag{3.9}$$

whose first-order optimality conditions are

$$Q_r^\top \widehat{c} + \frac{2\sigma \Lambda_r^{-1} z_r^*}{1 - (z_r^*)^\top \Lambda_r^{-1} z_r^*} = 0, \text{ and } 1 - (z_r^*)^\top \Lambda_r^{-1} z_r^* > 0. \tag{3.10}$$

Let $\theta = 1 - (z_r^*)^\top \Lambda_r^{-1} z_r^*$. Then equation (3.10) implies that $z_r^* = -\frac{\theta \Lambda_r Q_r^\top \widehat{c}}{2\sigma}$. Substituting this expression for z_r^* into the definition of θ, we obtain $\theta^2 \frac{\gamma}{4\sigma^2} + \theta - 1 = 0$, which has a positive root $\theta = \frac{2\sigma\sqrt{\sigma^2+\gamma} - 2\sigma^2}{\gamma}$. Hence, $y^* = -\frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma} B\widehat{c}$. Since

$$1 - (z_r^*)^\top \Lambda_r^{-1} z_r^* = 1 - (y^*)^\top B^\dagger y^* = 1 - \frac{\left(\sqrt{\sigma^2+\gamma} - \sigma\right)^2}{\gamma} = \frac{2\sigma\sqrt{\sigma^2+\gamma} - 2\sigma^2}{\gamma} > 0,$$

and $\nabla^2 \phi_\sigma(y) \succeq 0$, y^* is an optimal solution of (3.7). Furthermore, problems (3.2) and (3.7) are equivalent if

$$\frac{\sqrt{\sigma^2+\gamma} - \sigma}{\gamma} = \sqrt{\frac{1-\nu}{\gamma}},$$

that is $\nu = 2\sigma \frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$. □

REMARK 3.3. *Note from* (3.8) *that* $\lim_{\sigma \to 0} y = -\frac{B\widehat{c}}{\sqrt{\gamma}}$.

We now consider the logarithmic barrier problem for (3.1), i.e.,

$$\begin{aligned} \min_{X \in S^n} \quad & \phi_\sigma(X) := \langle C, X \rangle - \sigma \log \det X \\ \text{s.t.} \quad & X_{ii} = 1, \forall i = 1, \cdots, n, \quad X \succeq 0, \end{aligned} \tag{3.11}$$

where we define $\log\det(X)$ to be negative infinity for X not positive definite. Given a row i and fixing the block $B = X_{i^c,i^c}$, we have from (1.2) that

$$\det(X) = \det(B)(1 - X_{i^c,i}^\top B^{-1} X_{i^c,i}),$$

which implies that

$$\phi_\sigma(X) := \widehat{c}^\top X_{i^c,i} - \sigma \log(1 - X_{i^c,i}^\top B^{-1} X_{i^c,i}) + w(B),$$

where $\widehat{c} = 2C_{i^c,i}$ and $w(B)$ is a function of B (i.e., a constant). Hence, problem (3.11) becomes the unconstrained minimization problem

$$\min_{y \in \mathbb{R}^{n-1}} \quad \widehat{c}^\top y - \sigma \log(1 - y^\top B^{-1} y), \tag{3.12}$$

which is a special version of problem (3.7). Therefore, it follows from Lemma 3.2 that Algorithm 2 is essentially a row-by-row method for solving problem (3.11), if in that algorithm ν is replaced by $2\sigma \frac{\sqrt{\sigma^2+\gamma}-\sigma}{\gamma}$.

Our RBR method can be extended to solve

$$\begin{aligned} \min_{X \in S^n} \quad & \psi_\sigma(X) := f(X) - \sigma \log\det X \\ \text{s.t.} \quad & X \in \mathcal{X} := \{X \in S^n \mid L \le X \le U, X \succeq 0\}. \end{aligned} \tag{3.13}$$

where $f(X)$ is a convex function of X, the constant matrices $L, U \in S^n$ satisfy $L \le U$ and $L \le X$ means that $L_{i,j} \le X_{i,j}$ for all $i, j = 1, \cdots, n$. Note that problem (3.13) includes problem (3.11) as a special case if $L_{i,i} = U_{i,i} = 1$ for $i = 1, \cdots, n$ and $L_{i,j} = -\infty$ and $U_{i,j} = \infty$, otherwise. Starting from the point $X^k \succ 0$ at the k-th cycle, we fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{i^c,i^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{i,i}$ and $X_{i^c,i}$ (i.e., row i/column i), respectively; i.e., $X^k :\approx \begin{pmatrix} \xi & y^\top \\ y & B \end{pmatrix}$. Hence, the RBR subproblem of (3.13) becomes

$$\begin{aligned} \min_{X \in S^n} \quad & \widetilde{f}(\xi, y) - \sigma \log(\xi - y^\top B^{-1} y) \\ \text{s.t.} \quad & \begin{pmatrix} L_{i,i} \\ L_{i^c,i} \end{pmatrix} \le \begin{pmatrix} \xi \\ y \end{pmatrix} \le \begin{pmatrix} U_{i,i} \\ U_{i^c,i} \end{pmatrix}, \end{aligned} \tag{3.14}$$

where $\widetilde{f}(\xi, y) := f(X^k)$. Inspired by Proposition 2.7.1 in [6], we now prove a basic convergence result for the RBR method applied to problem (3.13).

Theorem 3.4. *Let $\{X^k\}$ be a sequence generated by the row-by-row method for solving* (3.14)*. Then every limit point of $\{X^k\}$ is a global minimizer of* (3.14)*.*

Proof. Clearly, our RBR method produces a sequence of nondecreasing objective function values

$$\psi_\sigma(X^k) \ge \psi_\sigma(X^{k,1}) \ge \psi_\sigma(X^{k,2}) \ge \cdots \ge \psi_\sigma(X^{k,n-1}) \ge \psi_\sigma(X^{k+1}). \tag{3.15}$$

Let $\widetilde{X}$ be a limit point of the sequence $\{X^k\}$. It follows from equation (3.15) that the sequences $\{\psi_\sigma(X^k)\}$, $\{\psi_\sigma(X^{k,1})\}$, $\cdots$, $\{\psi_\sigma(X^{k,n-1})\}$ all converge to $\psi_\sigma(\widetilde{X})$. Hence, $\widetilde{X}$ must be postitive definite because $\widetilde{X}$ is the limit point of a sequence of matrices that all lie in the compact level set $\{X \in \mathcal{X} | \psi_\sigma(X) \le \psi_\sigma(X^1)\}$, all of whose members are positive definite. We now show that $\widetilde{X}$ minimizes $\psi_\sigma(X)$.

Let $\{X^{k_j}\}$ be a subsequence of $\{X^k\}$ that converges to $\widetilde{X}$. We first show that $\{X^{k_j,1} - X^{k_j}\}$ converges to zero as $j \to \infty$. Assume on the contrary, that $\{X^{k_j,1} - X^{k_j}\}$ does not converges to zero. Then there exists a subsequence $\{\bar{k}_j\}$ of $\{k_j\}$ and some $\bar{\gamma} > 0$ such that $\gamma^{\bar{k}_j} := \|X^{\bar{k}_j,1} - X^{\bar{k}_j}\|_F \ge \bar{\gamma}$ for all j. Let $S^{\bar{k}_j,1} := (X^{\bar{k}_j,1} - X^{\bar{k}_j})/\gamma^{\bar{k}_j}$. Thus $X^{\bar{k}_j,1} = X^{\bar{k}_j} + \gamma^{\bar{k}_j} S^{\bar{k}_j,1}$, $\|S^{\bar{k}_j,1}\|_F = 1$ and $S^{\bar{k}_j,1}$ differs from zero only along the first row/column. Since $S^{\bar{k}_j,1}$ belongs to a compact set, it has a limit point $\bar{S}^1$. Hence, there exists a subsequence of $\{\hat{k}_j\}$ of $\{\bar{k}_j\}$ such that $S^{\hat{k}_j,1}$ converges to $\bar{S}^1$. Consider an arbitrary $t \in [0,1]$. Since $0 \le t\bar{\gamma} \le \gamma^{\hat{k}_j}$, $X^{\hat{k}_j} + tS^{\hat{k}_j,1}$ lies on the segment joining $X^{\hat{k}_j}$ and $X^{\hat{k}_j} + \gamma^{\hat{k}_j} S^{\hat{k}_j,1} = X^{\hat{k}_j,1}$, and belongs to $\mathcal{X}$ since $\mathcal{X}$ is a convex set. Moreover, since $X^{\hat{k}_j,1}$ uniquely minimizes $\psi_\sigma(X)$ over all X that differ from $X^{\hat{k}_j}$ along the first row/column, it follows from the convexity of $\psi_\sigma(X)$ that

$$\psi_\sigma(X^{\hat{k}_j,1}) = \psi_\sigma(X^{\hat{k}_j} + \gamma^{\hat{k}_j} S^{\hat{k}_j,1}) \le \psi_\sigma(X^{\hat{k}_j} + t\gamma^{\hat{k}_j} S^{\hat{k}_j,1}) \le \psi_\sigma(X^{\hat{k}_j}). \tag{3.16}$$

Since $\psi_\sigma(X^{\hat{k}_j,1})$ converges to $\psi_\sigma(\widetilde{X})$, it follows (3.16) that $\psi_\sigma(\widetilde{X}) \le \psi_\sigma(\widetilde{X} + t\bar{\gamma}\bar{S}^1) \le \psi_\sigma(\widetilde{X})$, which implies that $\psi_\sigma(\widetilde{X}) = \psi_\sigma(\widetilde{X} + t\bar{\gamma}\bar{S}^1)$ for all $t \in [0,1]$. Since $\bar{\gamma}\bar{S}^1 \ne 0$, this contradicts the fact that $\psi_\sigma(X)$ is strictly convex; hence $X^{k_j,1} - X^{k_j}$ converges to zero and $X^{k_j,1}$ converges to $\widetilde{X}$.

From the definition (3.13), we have

$$\psi_\sigma(X^{k_j,1}) \le \psi_\sigma(X), \quad \forall X \in V^{k_j,1} := \left\{ \begin{pmatrix} \xi & y^\top \\ y & X^{k_j}_{1^c,1^c} \end{pmatrix} \middle| \begin{pmatrix} \xi \\ y \end{pmatrix} \in \mathbb{R}^n, \begin{pmatrix} L_{1,1} \\ L_{1^c,1} \end{pmatrix} \le \begin{pmatrix} \xi \\ y \end{pmatrix} \le \begin{pmatrix} U_{1,1} \\ U_{1^c,1} \end{pmatrix} \right\}.$$

Taking the limit as j tends to infinity, we obtain that

$$\psi_\sigma(\widetilde{X}) \le \psi_\sigma(X), \quad \forall X \in V^1 := \left\{ \begin{pmatrix} \xi & y^\top \\ y & \widetilde{X}_{1^c,1^c} \end{pmatrix} \middle| \begin{pmatrix} \xi \\ y \end{pmatrix} \in \mathbb{R}^n, \begin{pmatrix} L_{1,1} \\ L_{1^c,1} \end{pmatrix} \le \begin{pmatrix} \xi \\ y \end{pmatrix} \le \begin{pmatrix} U_{1,1} \\ U_{1^c,1} \end{pmatrix} \right\},$$

which implies that, for any $p \in \{1, \cdots, n\}$,

$$\psi_\sigma(\widetilde{X}) \le \psi_\sigma(X), \quad \forall X \in V^1 \text{ and } X_{p^c,1} = \widetilde{X}_{p^c,1},$$

i.e., all components of the first row and column $[\xi; y]$ other than the p-th are fixed. Since $\widetilde{X}$ lies in the open convex set S^n_{++}, we obtain from the optimality conditions that, for any $p \in \{1, \cdots, n\}$,

$$\left\langle \nabla\psi_\sigma(\widetilde{X}), X - \widetilde{X} \right\rangle \ge 0, \quad \forall X \in V^1 \text{ and } X_{p^c,1} = \widetilde{X}_{p^c,1},$$

which further gives that, for any $p \in \{1, \cdots, n\}$,

$$\left(\nabla\psi_\sigma(\widetilde{X})\right)_{p,1} \left(X_{p,1} - \widetilde{X}_{p,1}\right) \ge 0, \quad \forall X_{p,1} \text{ such that } L_{p,1} \le X_{p,1} \le U_{p,1}. \tag{3.17}$$

Repeating the above argument shows that for $i = 2, \cdots, n$, the points $X^{k_j,i}$ also converges to $\widetilde{X}$ and

$$\left(\nabla\psi_\sigma(\widetilde{X})\right)_{p,i} \left(X_{p,i} - \widetilde{X}_{p,i}\right) \ge 0, \quad \forall L_{p,i} \le X_{p,i} \le U_{p,i}, \tag{3.18}$$

for any $p \in \{1, \cdots, n\}$. Therefore, for any $X \in \mathcal{X}$, it follows from (3.17) and (3.18) that

$$\left\langle \nabla\psi_\sigma(\widetilde{X}), X - \widetilde{X} \right\rangle = \sum_{i,j=1,\cdots,n} \left(\nabla\psi_\sigma(\widetilde{X})\right)_{i,j} \left(X_{i,j} - \widetilde{X}_{i,j}\right) \ge 0,$$

which implies that $\widetilde{X}$ is a global minimizer. □

4. A RBR method for SDP with general linear constraints.

We now consider SDP with general linear constraints. Unfortunately, in this case, the RBR method may not converge to an optimal solution of problem (2.1). This is similar to the fact that (block) coordinate descent method may not converge to an optimal solution for a linearly constrained convex problem [16]. It has long been known in [22] that the coordinate descent method for general nonlinear programming may not converge. Here is a 2-dimensional example that shows that for general linear constraints the RBR method may not converge to a global minimizer. Consider the SDP

$$\begin{aligned} \min \quad & X_{11} + X_{22} - \log\det(X) \\ \text{s.t.} \quad & X_{11} + X_{22} \ge 4, \quad X \succeq 0. \end{aligned} \tag{4.1}$$

Starting from a point X, where $X_{11} = 1$, $X_{12} = 0$ and $X_{22} = 3$, the RBR subproblems are

$$\min \quad X_{11} - \log(3X_{11} - X_{12}^2), \text{ s.t. } X_{11} \geq 1,$$

and

$$\min \quad X_{22} - \log(X_{22} - X_{12}^2), \text{ s.t. } X_{22} \geq 3,$$

since $\log\det(X) = \log(X_{11}X_{22} - X_{12}^2)$. It is readily verified that optimal solutions to these subproblems are, respectively, $X_{11} = 1$, $X_{12} = 0$ and $X_{12} = 0$, $X_{22} = 3$; hence, the row-by-row method remains at the initial point, while the true optimal solution is $X = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$.

To overcome this type of failure, the coordinate descent method is usually applied to a sequence of unconstrained problems obtained by penalizing the constraints in the objective function. We adopt a similar approach here by embedding the pure RBR method in an augmented Lagrangian function framework. We then introduce specialized versions of this algorithm for the SDP relaxation of the maxcut problem (3.1) and the minimum nuclear norm matrix completion problem.

4.1. A RBR augmented Lagrangian method. In this subsection, we first introduce an augmented Lagrangian method and then combine it with the row-by-row method for solving the standard form SDP (2.1).

The augmented Lagrangian function for problem (2.1) taking into consideration only the general linear constraints $\mathcal{A}(X) = b$ is defined as:

$$\mathcal{L}(X, \pi, \mu) := \langle C, X \rangle - \pi^\top (\mathcal{A}(X) - b) + \frac{1}{2\mu} \|\mathcal{A}(X) - b\|_2^2, \tag{4.2}$$

where $\pi \in \mathbb{R}^m$ and $\mu > 0$. Starting from $\pi^1 = \mathbf{0}$ and $\mu^1 \in (0, +\infty)$, our augmented Lagrangian method iteratively solves

$$X^k := \arg\min_X \mathcal{L}(X, \pi^k, \mu^k), \qquad \text{s.t. } X \succeq 0, \tag{4.3}$$

chooses $\mu^{k+1} \in [\gamma\mu^k, \mu^k]$ and then updates the vector of Lagrange multipliers by

$$\pi^{k+1} := \pi^k - \frac{\mathcal{A}(X^k) - b}{\mu^k}, \tag{4.4}$$

for the next iteration $k + 1$. It is important to note that our algorithm does not incorporate the positive semidefinite constraint into the augmented Lagrangian function, and therefore, it is different from the methods in [21, 30].

As is well known (see chapter 12.2 in [10]), (4.3) is equivalent to minimizing a quadratic penalty function:

$$X^k := \arg\min_X \ \mathcal{F}(X, b^k, \mu^k) := \langle C, X \rangle + \frac{1}{2\mu^k} \|\mathcal{A}(X) - b^k\|_2^2, \text{ s.t. } X \succeq 0, \tag{4.5}$$

where $b^k = b + \mu^k \pi^k$ and the difference between $\mathcal{L}(X, \pi^k, \mu^k)$ and $\mathcal{F}(X, b^k, \mu^k)$ is the constant $-\frac{\mu^k}{2}\|\pi^k\|_2^2$. Hence, we consider an alternative version of the augmented Lagrangian method which solves (4.5) and updates b^k by

$$b^{k+1} := b + \frac{\mu^{k+1}}{\mu^k}\left(b^k - \mathcal{A}(X^k)\right), \tag{4.6}$$

where $b^1 := b$. We now apply the RBR method to minimize (4.5). Starting from the point $X^k \succ 0$ at the k-th iteration, the RBR subproblem corresponding to the quadratic SDP (4.5) that is obtained by fixing all elements of X^k other than those in the i-th row and column results in a minimization problem with two conic constraints. Specifically, we fix the $n(n-1)/2$ variables in the $(n-1)\times(n-1)$ submatrix $B := X^k_{i^c,i^c}$ of X^k and let ξ and y denote the remaining unknown variables $X_{i,i}$ and $X_{i^c,i}$ (i.e., row i/column i), respectively. Hence, the quadratic SDP problem (4.5) becomes, after, replacing the zero on the right hand side of the Schur complement constraints by $\nu > 0$ to ensure positive definiteness of X^k,

$$\begin{aligned} \min_{(\xi;y)\in\mathbb{R}^n} \quad & \widetilde{c}^\top \begin{pmatrix}\xi \\ y\end{pmatrix} + \frac{1}{2\mu^k}\left\|\widetilde{A}\begin{pmatrix}\xi \\ y\end{pmatrix} - \widetilde{b}\right\|_2^2 \\ \text{s.t.} \quad & \xi - y^\top B^{-1} y \geq \nu, \end{aligned} \tag{4.7}$$

where $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ are given by (2.5) with b_i for $i = 1, \cdots, m$ replaced by b_i^k. If we let $LL^\top = B$ be the Cholesky factorization of B and introduce a new variable $z = L^{-1}y$, problem (4.7) can be written as:

$$\begin{aligned} \min_{(\xi;z;\tau)} \quad & \widetilde{c}^\top \begin{pmatrix}\xi \\ Lz\end{pmatrix} + \frac{1}{2\mu}\tau \\ \text{s.t.} \quad & \left\|\widetilde{A}\begin{pmatrix}\xi \\ Lz\end{pmatrix} - \widetilde{b}\right\|_2^2 \leq \tau \\ & \|z\|_2^2 \leq \xi - \nu. \end{aligned} \tag{4.8}$$

Therefore, each step of our RBR augmented Lagrangian method involves solving a SOCP with two rotated second-order cone constraints. We plan to show how advantage can be taken of the particular form of these SOCPs in a future paper. If B is only positive semidefinite, we can derive a similar SOCP by using the spectral decomposition of B. For references on solving SOCPs, see [1] for example. Our combined RBR augmented Lagrangian method for minimizing (2.1) is presented in Algorithm 3.

The RBR method applied to problem (4.5) converges by Theorem 3.4 since solving the RBR subproblem (4.7) essentially corresponds to minimizing the unconstrained function obtained by subtracting $\sigma \log(\xi - y^\top B^{-1} y)$ from the objective function in (4.7) using an argument analogous to the one made in section 3.1. Moreover, the convergence of our augmented Lagrangian framework follows from the standard theory for the augmented Lagrangian method for minimizing a strictly convex function subject to linear equality constraints [4, 5, 23].

4.2. Application to SDPs with only diagonal element constraints. Since the constraints in problem (3.1) are $X_{i,i} = 1$ for $i = 1, \cdots, n$, the quadratic term in the objective function of the RBR subproblem simplifies to

$$\left\|\widetilde{A}\begin{pmatrix}\xi \\ y\end{pmatrix} - \widetilde{b}\right\|^2 = (\xi - b_i^k)^2,$$

Algorithm 3: Row-by-row augmented Lagrangian method

```
    Set $X^1 \succ 0$, $b^1 = b$, $\eta \in (0,1)$, $\nu > 0$, $\mu^1 > 0$, $\epsilon, \epsilon_r, \epsilon_f \geq 0$ and $k := 1$.
    Compute $F^0 := \langle C, X^1 \rangle$ and set $F^1 := +\infty$.
    while $\frac{F^{k-1}-F^k}{\max\{|F^{k-1}|,1\}} \geq \epsilon$ or $\|\mathcal{A}(X^k) - b\|_2 \geq \epsilon_r$ do
        Compute $f^0 := \langle C, X^k \rangle + \frac{1}{2\mu^k}\|\mathcal{A}(X^k) - b^k\|_2^2$ and set $f^1 := +\infty$.
        while $\frac{f^{k-1}-f^k}{\max\{|f^{k-1}|,1\}} \geq \epsilon_f$ do
            for $i = 1, \cdots, n$ do
S1              Set $B := X^k_{i^c,i^c}$ and compute $\widetilde{c}$, $\widetilde{A}$ and $\widetilde{b}$ from (2.5) with $b$ replaced by $b^k$.
S2              Solve the SOCP (4.7) and denote its solution by $\xi$ and $y$.
S3              Set $X^k_{i,i} := \xi$, $X^k_{i^c,i} := y$ and $X^k_{i,i^c} := y^\top$.
            Compute $F^k := \langle C, X^k \rangle$ and $f^k := F^k + \frac{1}{2\mu^k}\|\mathcal{A}(X^k) - b^k\|_2^2$.
S4      Update $b^{k+1} := b + \frac{\mu^{k+1}}{\mu^k}\left(b^k - \mathcal{A}(X^k)\right)$.
        Choose $\mu^{k+1} \in [\eta\mu^k, \mu^k]$ and set $X^{k+1} := X^k$ and $k := k+1$.
```

and problem (4.7) reduces to

$$\begin{aligned} \min_{(\xi;y)\in\mathbb{R}^n} \quad & c\xi + \widehat{c}^\top y + \frac{1}{2\mu^k}(\xi - b_i^k)^2 \\ \text{s.t.} \quad & \xi - y^\top B^{-1} y \geq \nu, \end{aligned} \tag{4.9}$$

where $c := C_{i,i}$, $\widehat{c} := 2C_{i^c,i}$ and $b_i^1 = 1$. The first-order optimality conditions for (4.9) are

$$\begin{aligned} &\xi = b_i^k + \mu^k(\lambda - c) \\ &y = -\frac{1}{2\lambda} B\widehat{c} \\ &\xi \geq y^\top B^{-1} y + \nu, \quad \lambda \geq 0 \text{ and } (\xi - y^\top B^{-1} y - \nu)\lambda = 0. \end{aligned}$$

If $\widehat{c} = 0$, then $y = 0$ and $\xi = \max\{\nu, b_i^k - \mu^k c\}$. Otherwise, λ is the unique real root of the cubic equation:

$$\varphi(\lambda) := 4\mu^k\lambda^3 + 4(b_i^k - \mu^k c - \nu)\lambda^2 - \gamma = 0, \tag{4.10}$$

which is positive. This follows from the continuity of $\varphi(\lambda)$ and the facts that $\varphi(0) = -\widehat{c}^\top B\widehat{c} < 0$, $\lim_{\lambda\to+\infty}\varphi(\lambda) = +\infty$ and

$$\varphi'(\lambda) = 12\mu^k\lambda^2 + 8(b_i^k - \mu^k c - \nu)\lambda \geq 4\mu^k\lambda^2$$

since $\xi = b_i^k - \mu^k c + \mu^k\lambda \geq \nu$, which implies that $\varphi'(0) = 0$ and $\varphi'(\lambda) > 0$ for $\lambda \neq 0$. We now present a specialized version of the RBR augmented Lagrangian method for problem (3.1) as Algorithm 4.

4.3. Matrix Completion. Given a matrix $M \in \mathbb{R}^{p\times q}$ and an index set

$$\Omega \subseteq \{(i,j) \mid i \in \{1,\cdots,p\}, j \in \{1,\cdots,q\}\},$$

Algorithm 4: Row-by-row augmented Lagrangian method (ALAG-RBR-M) for problem (3.1)

```
This is a specialized version of Algorithm 3. In Algorithm 3,
1. replace $\mathcal{A}(X^k)$ by $\mathrm{diag}(X^k)$;
2. replace steps S1-S3 by the following steps:
   Set $c := C_{i,i}$, $\widehat{c} := 2C_{i^c,i}$ and $B := X^k_{i^c,i^c}$.
   If $\widehat{c} = 0$, set $X^k_{ii} = \max\{\nu, b^k_i - \mu^k c\}$, $X^k_{i^c,i} = (X^k_{i,i^c})^\top = 0$.
   Otherwise, compute the positive solution $\lambda$ of (4.10), and set $X^k_{i,i} = b^k_i + \mu^k(\lambda - c)$, $X^k_{i^c,i} = -\frac{1}{2\lambda}B\widehat{c}$
   and $X^k_{i,i^c} = (X^k_{i^c,i})^\top$.
```

the nuclear norm matrix completion problem is

$$\begin{array}{lll} (4.11) & \min_{W\in\mathbb{R}^{p\times q}} & \|W\|_* \\ & s.t. & W_{ij} = M_{ij},\ \forall\,(i,j)\in\Omega. \end{array}$$

An equivalent SDP formulation of (4.11) is

$$\begin{array}{lll} & \min_{X\in S^n} & Tr(X) \\ (4.12) & s.t. & X := \begin{bmatrix} X^{(1)} & W \\ W^\top & X^{(2)} \end{bmatrix} \succeq 0 \\ & & W_{ij} = M_{ij},\ \forall\,(i,j)\in\Omega, \end{array}$$

where $n = p + q$ and the number of linear constraints is $m = |\Omega|$. Let M_Ω be the vector whose elements are the components of $\{M_{i,j} \mid (i,j)\in\Omega\}$ obtained by stacking the columns of M from column 1 to column q and then keeping only those elements that are in Ω. Hence, M_Ω corresponds to the right hand side b of the constraints in the general SDP (2.1).

We now present the RBR subproblem (4.7) corresponding to problem (4.12). First, the vector y can be partitioned into two subvectors whose elements are, respectively, in and not in the set Ω:

$$y :\approx \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix}, \quad \widehat{y} := X_{\alpha,i}, \text{ and } \widetilde{y} := X_{\beta,i},$$

where, the index sets α and β are

$$(4.13) \qquad \beta := i^c\backslash\alpha, \quad \alpha := \begin{cases} \{j+p, \mid j\in\bar{\alpha}\}, \text{ where } \bar{\alpha} := \{j \mid (i,j)\in\Omega, j = 1,\cdots,q\}, \text{ if } i\le p, \\ \{j \mid (j,i)\in\Omega, j = 1,\cdots,p\}, \text{ if } p < i \le n. \end{cases}$$

Hence, it follows from the special constraints $W_{ij} = M_{ij}, (i,j)\in\Omega$ in (4.11) that the norm of the residual of each RBR subproblem (4.7) is

$$\left\|\widetilde{A}\begin{pmatrix}\xi \\ y\end{pmatrix} - \widetilde{b}\right\| = \left\|X_{\alpha,i} - \widetilde{b}\right\| =: \|\widehat{y} - \widetilde{b}\|,$$

where

$$(4.14) \qquad \widetilde{b} := \begin{cases} (M^k_{i,\bar{\alpha}})^\top, & \text{if } i\le p, \\ M^k_{\alpha,i-p}, & \text{if } p < i \le n, \end{cases}$$

and $M^1 = M$. Therefore, the SOCP (4.7) becomes

$$
\begin{aligned}
\min_{(\xi;y)\in\mathbb{R}^n} \quad & \xi + \frac{1}{2\mu^k}\left\|\widehat{y} - \widetilde{b}\right\|_2^2 \\
\text{s.t.} \quad & \xi - y^\top B^{-1} y \geq \nu,
\end{aligned}
\tag{4.15}
$$

where the matrix $B = \begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}$.

LEMMA 4.1. *The optimal solution of the RBR subproblem*(4.15) *is given by*

$$
\begin{cases}
\xi = \dfrac{1}{2\mu^k}\widehat{y}^\top(\widetilde{b} - \widehat{y}) + \nu, \\
\widehat{y} = \left(2\mu^k I + X^k_{\alpha,\alpha}\right)^{-1} X^k_{\alpha,\alpha}\widetilde{b}, \quad \widetilde{y} = \dfrac{1}{2\mu^k} X^k_{\beta,\alpha}(\widetilde{b} - \widehat{y}).
\end{cases}
\tag{4.16}
$$

Proof. Note that the optimal solution $[\xi; y] = [\xi; \widehat{y}; \widetilde{y}]$ of (4.15) must satisfy $\xi = y^\top B^{-1} y + \nu$. Hence, (4.15) is equivalent to an unconstrained quadratic minimization problem

$$
\min_y y^\top B^{-1} y + \frac{1}{2\mu^k}\left\|\widehat{y} - \widetilde{b}\right\|_2^2,
\tag{4.17}
$$

whose optimality conditions are

$$
\begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} + \frac{1}{2\mu^k}\begin{pmatrix} \widehat{y} - \widetilde{b} \\ \mathbf{0} \end{pmatrix} = 0,
\tag{4.18}
$$

which implies that

$$
\begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} + \frac{1}{2\mu^k}\begin{pmatrix} X^k_{\alpha,\alpha} \\ X^k_{\beta,\alpha} \end{pmatrix}\widehat{y} = \frac{1}{2\mu^k}\begin{pmatrix} X^k_{\alpha,\alpha} \\ X^k_{\beta,\alpha} \end{pmatrix}\widetilde{b}.
$$

Therefore, $\widetilde{y} = \frac{1}{2\mu^k} X^k_{\beta,\alpha}(\widetilde{b} - \widehat{y})$, where $\widehat{y}$ can be computed from the system of linear equations

$$
\left(2\mu^k I + X^k_{\alpha,\alpha}\right)\widehat{y} = X^k_{\alpha,\alpha}\widetilde{b}.
$$

Then, it follows from $\xi = y^\top B^{-1} y + \nu$ and (4.18) that

$$
\xi = \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix}^\top \begin{pmatrix} X^k_{\alpha,\alpha} & X^k_{\alpha,\beta} \\ X^k_{\beta,\alpha} & X^k_{\beta,\beta} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix} + \nu = \frac{1}{2\mu^k}\begin{pmatrix} \widehat{y} \\ \widetilde{y} \end{pmatrix}^\top \begin{pmatrix} \widetilde{b} - \widehat{y} \\ \mathbf{0} \end{pmatrix} + \nu = \frac{1}{2\mu^k}\widehat{y}^\top(\widetilde{b} - \widehat{y}) + \nu.
\tag{4.19}
$$

□

Note from (4.16) that we only need to solve a single system of linear equations, whose size is the number of sample elements in the row and hence expected to be small, to obtain the minimizer of the RBR subproblem (4.15). The specialized RBR method for minimizing (4.12) is presented in Algorithm 5.

5. Numerical Results.

Although the numerical results that we present in this section are limited to two special classes of SDP problems, they illustrate the effectiveness of our RBR algorithmic framework. Specifically, they show that large scale SDPs can be solved in a moderate amount of time using only moderate

Algorithm 5: The RBR method (RBR-MC) for problem (4.12)

```
Set $X^1 \succ 0$, $b^1 = b$, $\eta \in (0,1)$, $\nu > 0$, $\mu^1 > 0$, $\epsilon, \epsilon_r, \epsilon_f \geq 0$ and $k := 1$.
Compute $F^0 := Tr(X^1)$ and set $F^1 := +\infty$.
while $\frac{F^{k-1}-F^k}{\max\{|F^{k-1}|,1\}} \geq \epsilon$ or $\|X^k_\Omega - M_\Omega\|_2 \geq \epsilon_r$ do
    Compute $f^0 := Tr(X^k) + \frac{1}{2\mu^k}\|X^k_\Omega - M^k_\Omega\|_2^2$ and set $f^1 := +\infty$.
    while $\frac{f^{k-1}-f^k}{\max\{|f^{k-1}|,1\}} \geq \epsilon_f$ do
        for $i = 1, \cdots, n$ do
            Set $\alpha$ and $\beta$ by (4.13), and $\widetilde{b}$ by (4.14).
            if $|\alpha| = 0$ then Set $X^k_{i,i} = 0$, $X^k_{i^c,i} = 0$ and $X^k_{i,i^c} = 0$.
            else
                Compute $X^k_{\alpha,i} := \left(2\mu^k I + X^k_{\alpha,\alpha}\right)^{-1} X_{\alpha,\alpha}\widetilde{b}$; $X^k_{\beta,i} = \frac{1}{2\mu} X_{\beta,\alpha}(\widetilde{b} - X^k_{\alpha,i})$;
                $X^k_{i,i} = \frac{1}{2\mu}(X^k_{\alpha,i})^\top(\widetilde{b} - X^k_{\alpha,i}) + \nu$ and set $X^k_{i,i^c} = (X^k_{i^c,i})^\top$.
        Compute $F^k := Tr(X^k)$ and $f^k := F^k + \frac{1}{2\mu^k}\|X^k_\Omega - M^k_\Omega\|_2^2$.
    Update $M^{k+1}_\Omega := M_\Omega + \frac{\mu^{k+1}}{\mu^k}\left(M^k_\Omega - X^k_\Omega\right)$.
    Choose $\mu^{k+1} \in [\eta\mu^k, \mu^k]$ and set $X^{k+1} := X^k$ and $k := k+1$.
```

amount of memory. Moreover, our tests show that the number cycles taken by our algorithm grows very slowly with the size of the problem.

5.1. The maxcut SDP relaxation. In this subsection, we demonstrate the effectiveness of the RBR methods Algorithms 2 (PURE-RBR-M) and 4 (ALAG-RBR-M) on a set of maxcut SDP relaxation problems and compare them with the code DSDP (version 5.8) [3]. The DSDP code implements a dual interior point method that is designed to take advantage of the structure of such problems. The main parts of our code were written in C Language MEX-files in MATLAB (Release 7.3.0), and all experiments were performed on a Dell Precision 670 workstation with an Intel Xeon 3.4GHZ CPU and 6GB of RAM.

The test problems are based on graphs generated by "rudy", a machine independent graph generator written by G.Rinaldi. These graphs range in size from $n = 1000$ to $n = 4000$ and the arguments of "rudy" are similar to those used in the G Set of graphs tested in [3, 17]. Specifically, for size $n = 1000$, we produced five different graphs "R1000-1", $\cdots$, "R1000-5" with a density of 1% (4,995 edges) and with random edge weights from $\{-1, 1\}$ by using the command

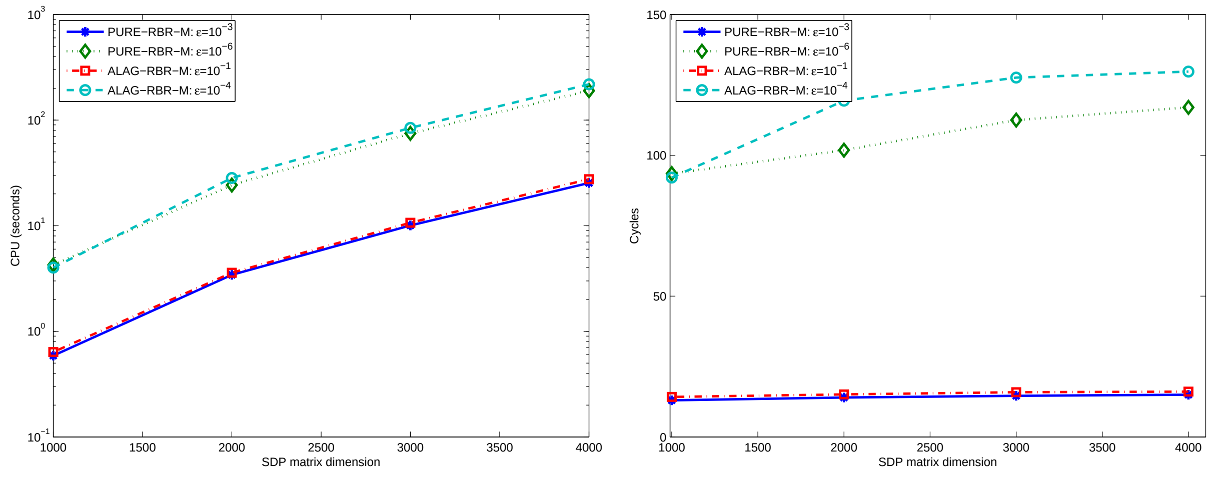
```
rudy -rnd_graph n 1 seed -random 0 1 seed  -times 2 -plus -1,
```

where seed $= 10n + i$, $i = 1, \cdots, 5$, respectively. The graphs from "R2000-1" to "R4000-5" were generated in a similar fashion. We also tested "almost" planar graphs having as their edge set the union of the edges of two (almost maximal) planar graphs with random edge weights from $\{-1, 1\}$; that is, the graphs "P1000-1" to "P4000-5" were generated by the command

```
rudy -planar n 99 seed1 -planar n  99 seed2  + -random 0 1 seed3 -times 2 -plus -1,
```

where seed1 $= 10n + i + 4$, seed2 $= 10n + i + 5$ and seed3 $= 10n + i$, for $n = 1000, \cdots, 4000$ and $i = 1, \cdots, 5$. In all cases the cost matrix C was the Laplace matrix of the graph divided by 4, i.e., $C = -\frac{1}{4}(\text{diag}(Ae) - A)$, where A was the weighted adjacency matrix of the graph.

The parameters of DSDP were set to their default values. The parameter ν in the RBR methods was set to 10^{-6}. We ran PURE-RBR-M with two different tolerances, i.e., ϵ was set to 10^{-3} (moderately accurate) and 10^{-6} (highly accurate), respectively. Similarly, we ran ALAG-RBR-M with two different tolerance settings, that is, $\epsilon, \epsilon_r, \epsilon_f$ were all set to 10^{-1} and 10^{-4}, respectively. For practical considerations, we



(a) CPU Time (b) Cycles

FIG. 5.1. *Relationship between the computational cost and SDP matrix dimension for the maxcut SDP relaxation*

terminated minimizing each augmented Lagrangian function if the number of cycles was greater than 5. The initial penalty parameter μ^1 in ALAG-RBR-M was set to 5 and was updated by $\mu^{k+1} = \max(0.5\mu^k, 10^{-1})$.

A summary of the computational results is presented in Table 5.1. In the table, “obj” denotes the objective function of the dual problem computed by DSDP, “rel-obj” denotes the relative error between the objective function value computed by the RBR methods and “obj”, “CPU” denotes CPU time measured in seconds, and “cycle” denotes the total number RBR cycles. From the table, we can see that both RBR methods are able to solve the maxcut SDP relaxation very efficiently. The number of cycles required was almost the same for all of the problems, no matter what their size was.

To illustrate the relationship between the computational cost of the RBR methods and the dimension of the SDP matrices, we plot the average of the CPU time versus the dimension in Figure 5.1 (a) and the average of the number of cycles versus the dimension in Figure 5.1 (b). Somewhat surprisingly, our augmented Lagrangian RBR algorithm solved the maxcut SDP problems as efficiently as our pure RBR algorithm for a given relative error.

5.2. Matrix Completion. In this subsection, we evaluate the RBR method Algorithm 5 (RBR-MC) on the matrix completion problem (4.12). Note that the pure RBR method can be directly applied to this problem. However, preliminary numerical testing showed that this approach is much slower (i.e., converges much more slowly) than using RBR-MC. Random matrices $M \in \mathbb{R}^{p\times q}$ with rank r were created by the following procedure [20]: we first generated random matrices $M_L \in \mathbb{R}^{p\times r}$ and $M_R \in \mathbb{R}^{q\times r}$ with i.i.d. Gaussian entries and then set $M = M_L M_R^\top$; then we sampled a subset Ω of m entries uniformly at random. The ratio $m/(pq)$ between the number of measurements and the number of entries in the matrix is denoted by “SR” (sampling ratio). The ratio $r(p+q-r)/m$ between the dimension of a rank r matrix to the number of samples is denoted by “FR”. In our tests, the rank r and the sampling entry m were taken consistently so that according to the theory in [9] the matrix M is the optimal solution of problem (4.12). Specifically, FR was set to 0.2 and 0.3 and r was set to 10. We tested five square matrices M with dimensions $p = q \in \{100, 200, \cdots, 500\}$ and set the number m to $r(p+q-r)/\mathrm{FR}$. All parameters p, q, r, m and the random seeds “seed” used by the random number generators “rand” and “randn” in MATLAB are reported in Table 5.2.

Table 5.1
Computational results for the maxcut SDP relaxation.

	DSDP		PURE-RBR-M						ALAG-RBR-M					
			$\epsilon = 10^{-3}$			$\epsilon = 10^{-6}$			$\epsilon = \epsilon_r = \epsilon_f = 10^{-1}$			$\epsilon = \epsilon_r = \epsilon_f = 10^{-4}$		
Name	obj	CPU	rel-obj	CPU	cycle	rel-obj	CPU	cycle	rel-obj	CPU	cycle	rel-obj	CPU	cycle
random graphs														
R1000-1	-1.4e+3	52.6	4.9e-3	0.6	13	3.0e-5	3.9	90	5.4e-3	0.6	13	3.2e-5	3.7	87
R1000-2	-1.4e+3	57.0	5.0e-3	0.6	13	3.6e-5	4.1	96	4.9e-3	0.6	14	4.2e-5	3.7	88
R1000-3	-1.5e+3	50.8	5.0e-3	0.6	13	3.8e-5	4.3	99	4.9e-3	0.6	14	4.0e-5	4.2	99
R1000-4	-1.4e+3	51.3	5.0e-3	0.6	13	3.2e-5	4.0	94	4.8e-3	0.6	14	3.3e-5	3.9	92
R1000-5	-1.5e+3	50.1	4.6e-3	0.6	13	3.5e-5	3.7	87	4.1e-3	0.6	14	3.6e-5	3.4	81
R2000-1	-4.1e+3	607.6	5.0e-3	3.9	14	3.7e-5	26.5	97	5.9e-3	3.8	14	1.9e-5	33.5	121
R2000-2	-4.1e+3	602.3	5.2e-3	3.9	14	3.6e-5	27.5	101	5.5e-3	4.2	15	1.9e-5	35.4	127
R2000-3	-4.2e+3	680.5	5.1e-3	4.3	14	3.4e-5	26.4	97	5.3e-3	4.2	15	1.6e-5	33.7	123
R2000-4	-4.2e+3	646.7	5.2e-3	3.9	14	3.2e-5	26.1	96	5.2e-3	4.2	15	1.4e-5	32.3	118
R2000-5	-4.1e+3	661.5	4.9e-3	3.9	14	3.9e-5	26.1	96	5.9e-3	3.8	14	2.0e-5	34.5	126
R3000-1	-7.7e+3	2576	5.0e-3	12.8	15	4.1e-5	90.0	103	5.1e-3	13.9	16	2.1e-5	110.8	127
R3000-2	-7.7e+3	2606	5.2e-3	13.2	15	3.7e-5	89.4	105	5.2e-3	14.1	16	2.1e-5	111.2	128
R3000-3	-7.9e+3	2530	5.0e-3	13.0	15	4.0e-5	91.4	107	5.1e-3	14.0	16	2.5e-5	109.8	127
R3000-4	-7.9e+3	2518	5.1e-3	13.6	15	4.0e-5	98.0	107	5.2e-3	13.9	16	2.3e-5	110.3	128
R3000-5	-7.7e+3	2514	5.3e-3	12.9	15	3.7e-5	91.8	107	5.4e-3	14.0	16	1.9e-5	109.8	128
R4000-1	-1.2e+4	6274	5.9e-3	36.5	15	4.0e-5	261.1	108	6.2e-3	39.0	16	2.4e-5	316.5	130
R4000-2	-1.2e+4	6310	5.7e-3	36.3	15	3.9e-5	265.7	108	6.0e-3	39.0	16	2.1e-5	313.5	130
R4000-3	-1.2e+4	6529	5.8e-3	36.0	15	4.1e-5	264.1	110	5.9e-3	39.5	16	2.5e-5	313.5	130
R4000-4	-1.2e+4	7018	5.8e-3	36.6	15	3.7e-5	261.3	108	6.1e-3	39.3	16	2.1e-5	315.3	130
R4000-5	-1.2e+4	5994	5.6e-3	36.7	15	3.8e-5	270.1	112	5.1e-3	41.6	17	2.5e-5	309.8	129
random planar graphs														
P1000-1	-1.4e+3	45.1	5.0e-3	0.6	13	4.0e-5	4.9	102	4.1e-3	0.7	15	3.8e-5	4.3	96
P1000-2	-1.4e+3	45.5	4.4e-3	0.6	13	2.9e-5	4.2	89	3.3e-3	0.6	14	2.3e-5	3.9	87
P1000-3	-1.5e+3	42.6	4.6e-3	0.6	13	3.5e-5	4.2	88	3.0e-3	0.7	15	2.7e-5	4.2	93
P1000-4	-1.4e+3	44.1	4.7e-3	0.6	13	3.8e-5	4.7	97	3.4e-3	0.7	14	3.8e-5	4.3	96
P1000-5	-1.4e+3	44.6	4.4e-3	0.6	13	3.2e-5	4.7	93	2.8e-3	0.7	15	2.4e-5	4.6	102
P2000-1	-2.9e+3	386.1	5.5e-3	3.0	14	3.7e-5	21.6	102	5.4e-3	3.0	15	2.5e-5	22.5	114
P2000-2	-2.8e+3	362.8	5.8e-3	2.9	14	3.9e-5	22.1	109	5.8e-3	3.2	16	2.9e-5	23.4	119
P2000-3	-2.9e+3	359.0	5.4e-3	2.9	14	3.7e-5	22.2	105	4.9e-3	3.2	16	2.5e-5	23.0	117
P2000-4	-2.9e+3	348.2	5.5e-3	2.9	14	4.0e-5	22.8	111	4.9e-3	3.0	15	2.9e-5	23.7	121
P2000-5	-2.9e+3	377.9	5.6e-3	2.9	14	3.9e-5	21.3	104	4.7e-3	3.2	16	3.2e-5	21.2	108
P3000-1	-4.3e+3	1400	6.0e-3	7.3	15	4.0e-5	56.3	117	6.2e-3	7.0	15	3.0e-5	58.3	127
P3000-2	-4.3e+3	1394	6.5e-3	7.0	14	4.7e-5	57.2	119	5.1e-3	7.5	16	3.3e-5	59.1	129
P3000-3	-4.4e+3	1351	6.3e-3	6.8	14	4.3e-5	57.0	118	5.2e-3	7.4	16	3.5e-5	58.1	128
P3000-4	-4.3e+3	1647	6.7e-3	7.0	14	4.8e-5	60.6	125	6.2e-3	7.4	16	4.0e-5	58.6	129
P3000-5	-4.3e+3	1757	6.7e-3	6.9	14	4.4e-5	57.0	117	5.7e-3	7.5	16	3.3e-5	57.2	125
P4000-1	-5.7e+3	3688	6.5e-3	14.3	15	4.3e-5	114.2	124	6.0e-3	15.5	16	3.0e-5	123.6	130
P4000-2	-5.9e+3	3253	6.5e-3	14.4	15	4.9e-5	116.7	126	6.2e-3	15.9	16	4.1e-5	125.2	130
P4000-3	-5.8e+3	3790	6.3e-3	14.2	15	4.8e-5	115.1	126	5.6e-3	15.3	16	3.8e-5	120.1	128
P4000-4	-5.8e+3	3474	6.8e-3	14.3	15	4.6e-5	118.8	128	6.5e-3	15.5	16	4.1e-5	123.8	131
P4000-5	-5.9e+3	3389	6.1e-3	14.3	15	4.4e-5	111.9	120	5.5e-3	15.4	16	3.6e-5	121.1	129

All parameters of RBR-MC were set to the same values as those used in ALAG-RBR-M. A summary of the computational results is presented in Table 5.2. In the table, "rel-X" denotes the relative error between the true and the recovered matrices, which is defined as rel-X $:= \frac{\|X-M\|_F}{\|M\|_F}$, and "rel-obj" denotes the relative error between the objective function value obtained and $\|M\|_*$, which is defined as rel-obj $:= \frac{|\frac{1}{2}Tr(X)-\|M\|_*|}{\|M\|_*}$. DSDP is not included in this comparison because it takes too long to solve all problems. To illustrate the relationship between the computational cost of the RBR methods and the dimension of the matrices, we plot the average of the CPU time versus the dimension of the SDP matrix (i.e., $p+q$) in Figure 5.2 (a) and the average of the number of cycles versus this dimension in Figure 5.2 (b).

6. Conclusion. In this paper, we present a new first-order algorithmic framework, the row-by-row (RBR) method, for solving semidefinite programming problems based on replacing the positive semidefinite

Table 5.2
Computational results for the matrix completion problem

	FR=0.2								FR=0.3							
	$\epsilon = 10^{-1}$				$\epsilon = 10^{-4}$				$\epsilon = 10^{-1}$				$\epsilon = 10^{-4}$			
seed	rel-X	rel-obj	CPU	cycle	rel-X	rel-obj	CPU	cycle	rel-X	rel-obj	CPU	cycle	rel-X	rel-obj	CPU	cycle
	p=q=100; r=10; m=9500; SR=0.95								p=q=100; r=10; m=6333; SR=0.63							
68521	1.7e-6	7.7e-3	0.5	8	3.3e-7	2.5e-4	2.8	42	5.1e-5	1.0e-3	0.3	10	4.3e-7	5.0e-5	1.7	51
56479	8.4e-7	3.7e-3	0.5	8	3.0e-7	2.9e-4	2.4	35	5.4e-5	4.9e-4	0.3	10	3.9e-7	5.0e-5	1.4	44
115727	1.3e-6	4.0e-3	0.5	8	3.4e-7	2.4e-4	2.4	37	3.8e-5	7.4e-4	0.3	10	4.5e-7	4.7e-5	1.4	43
27817	1.2e-6	4.6e-3	0.5	8	3.2e-7	2.4e-4	2.8	42	5.3e-5	1.0e-3	0.3	10	4.1e-7	5.4e-5	1.7	53
9601	1.1e-6	4.1e-3	0.6	8	3.1e-7	2.4e-4	2.7	40	3.1e-5	6.2e-4	0.3	10	4.1e-7	5.2e-5	1.6	49
	p=q=200; r=10; m=19500; SR=0.49								p=q=200; r=10; m=13000; SR=0.33							
68521	7.5e-5	1.1e-4	1.7	9	2.5e-7	6.4e-5	9.6	50	1.0e-3	4.9e-4	1.0	10	3.6e-7	1.9e-5	6.8	73
56479	6.0e-5	1.1e-4	1.8	9	2.3e-7	6.8e-5	8.1	42	1.5e-3	7.7e-4	0.9	10	3.3e-7	2.3e-5	8.1	87
115727	6.8e-5	2.1e-4	1.7	9	2.4e-7	6.8e-5	11.6	59	2.4e-3	6.0e-4	1.0	10	3.4e-7	2.0e-5	7.5	80
27817	5.3e-5	6.5e-4	1.7	9	2.3e-7	8.5e-5	14.0	74	2.5e-4	5.4e-4	0.9	10	3.2e-7	2.3e-5	7.8	84
9601	6.3e-5	3.3e-4	1.7	9	2.3e-7	8.0e-5	12.1	64	6.6e-4	5.0e-4	1.0	10	3.3e-7	2.1e-5	7.6	81
	p=q=300; r=10; m=29500; SR=0.33								p=q=300; r=10; m=19666; SR=0.22							
68521	1.0e-4	2.1e-4	3.3	9	2.0e-7	4.0e-5	21.7	59	1.0e-3	4.9e-4	2.0	11	3.0e-7	1.4e-5	17.7	96
56479	1.0e-4	2.8e-4	3.3	9	2.0e-7	3.8e-5	20.8	56	3.3e-4	4.3e-4	2.3	12	3.1e-7	1.3e-5	17.2	93
115727	9.4e-5	1.4e-4	3.3	9	2.0e-7	4.2e-5	24.7	67	1.2e-2	7.5e-4	2.4	13	3.2e-7	1.3e-5	15.5	83
27817	9.7e-5	7.1e-4	3.3	9	2.0e-7	3.7e-5	10.4	28	3.8e-3	5.0e-4	2.1	11	2.9e-7	1.3e-5	18.0	96
9601	1.0e-4	2.3e-3	3.3	9	1.9e-7	3.5e-5	9.5	26	1.8e-3	4.7e-4	2.2	12	2.9e-7	1.3e-5	17.2	93
	p=q=400; r=10; m=39500; SR=0.25								p=q=400; r=10; m=26333; SR=0.16							
68521	1.0e-4	1.2e-3	5.6	9	1.8e-7	2.6e-5	28.3	43	9.8e-3	6.2e-4	4.8	15	5.4e-6	9.2e-6	29.3	92
56479	9.9e-5	2.1e-4	5.8	9	1.8e-7	3.2e-5	48.1	71	3.0e-3	5.5e-4	4.5	14	2.7e-7	1.0e-5	31.8	101
115727	9.9e-5	1.0e-3	5.6	9	1.8e-7	2.7e-5	31.6	50	2.0e-3	5.0e-4	4.5	14	2.7e-7	1.0e-5	32.3	101
27817	2.2e-4	4.1e-4	5.8	9	1.8e-7	3.0e-5	37.9	57	8.3e-3	6.5e-4	4.5	14	2.8e-7	9.7e-6	32.6	100
9601	1.0e-4	1.3e-3	5.8	9	1.8e-7	2.5e-5	26.7	40	8.0e-3	5.5e-4	4.9	15	2.8e-7	9.6e-6	31.3	98
	p=q=500; r=10; m=49500; SR=0.20								p=q=500; r=10; m=33000; SR=0.13							
68521	2.4e-4	8.6e-4	9.1	9	1.7e-7	2.1e-5	56.0	56	5.3e-3	6.1e-4	8.2	16	2.6e-7	7.6e-6	54.4	107
56479	1.1e-4	8.5e-4	8.9	9	1.6e-7	2.3e-5	53.7	53	5.4e-3	5.9e-4	8.0	16	2.6e-7	7.2e-6	54.4	109
115727	1.1e-4	9.4e-4	9.4	9	1.6e-7	2.3e-5	49.8	48	8.2e-3	5.8e-4	8.1	16	2.6e-7	7.5e-6	54.4	108
27817	3.3e-4	1.5e-3	9.1	9	1.6e-7	2.1e-5	53.6	53	9.2e-3	6.7e-4	8.2	16	2.6e-7	7.6e-6	53.6	104
9601	1.1e-4	1.5e-3	9.9	9	1.6e-7	2.4e-5	54.9	53	2.1e-3	4.2e-4	8.0	16	2.5e-7	7.6e-6	57.3	111

constraint $X \succeq 0$ by requiring nonnegativity of the Schur complement of any $(n-1)$-dimensional principal submatrix of X, which is temporally fixed. By doing this, the positive semidefinite constraint is reduced to a simple second-order cone constraint. The pure RBR method is extremely effective in solving an SDP whose only constraints are that the diagonal elements of X are constant, since only a single matrix-vector product is involved at each inner step. To handle more general linear constraints in an SDP, we apply the pure RBR method to a sequence of unconstrained problems by using an augmented Lagrangian approach. Our method is especially suitable for solving the maxcut SDP relaxation and nuclear norm matrix completion problems since closed-form solutions for the RBR subproblems are available.

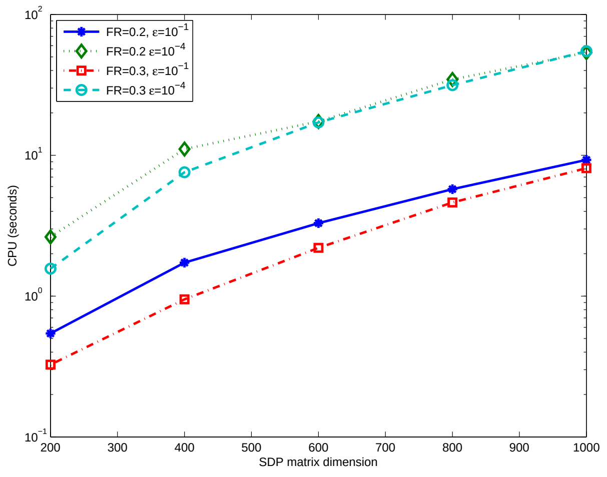
We intend to report in a follow-up paper on the extension of our RBR method to general conic programs that involve nonegativity and second-order cone constraints in addition to linear equality and positive semidefinite constraints.

REFERENCES

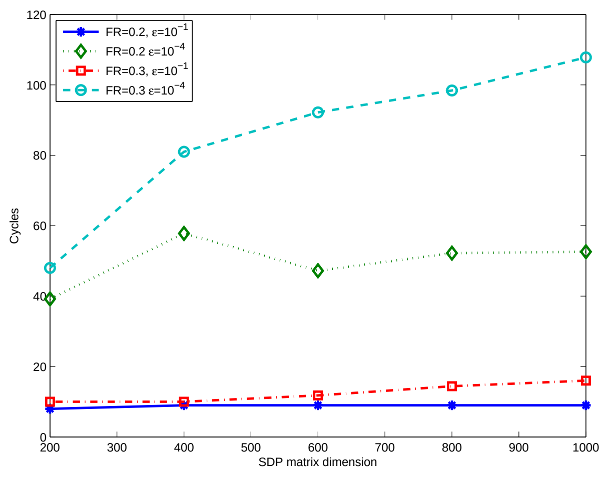
[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Math. Program., 95 (2003), pp. 3–51. ISMP 2000, Part 3 (Atlanta, GA).

[2] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, *Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data*, J. Mach. Learn. Res., 9 (2008), pp. 485–516.

[3] S. J. Benson, Y. Ye, and X. Zhang, *Solving large-scale sparse semidefinite programs for combinatorial optimization*, SIAM J. Optim., 10 (2000), pp. 443–461 (electronic).



(a) CPU Time



(b) Cycles

FIG. 5.2. *Relationship between the computational cost and SDP matrix dimension for SDP matrix completion*

[4] D. P. BERTSEKAS, *Necessary and sufficient condition for a penalty method to be exact*, Math. Programming, 9 (1975), pp. 87–99.

[5] ———, *Constrained optimization and Lagrange multiplier methods*, Computer Science and Applied Mathematics, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.

[6] ———, *Nonlinear Programming*, Athena Scientific, September 1999.

[7] S. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge University Press, Cambridge, 2004.

[8] S. BURER AND R. D. C. MONTEIRO, *A projected gradient algorithm for solving the maxcut SDP relaxation*, Optim. Methods Softw., 15 (2001), pp. 175–200.

[9] E. J. CANDES AND B. RECHT, *Exact matrix completion via convex optimization*, tech. rep., California Institute of Technology, May 2008.

[10] R. FLETCHER, *Practical methods of optimization*, A Wiley-Interscience Publication, John Wiley & Sons Ltd., Chichester, second ed., 1987.

[11] K. FUJISAWA, M. KOJIMA, AND K. NAKATA, *Exploiting sparsity in primal–dual interior-point methods for semidefinite programming*, Math. Programming, 79 (1997), pp. 235–253. Lectures on mathematical programming (ismp97) (Lausanne, 1997).

[12] M. X. GOEMANS AND D. P. WILLIAMSON, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42 (1995), pp. 1115–1145.

[13] D. GOLDFARB AND G. IYENGAR, *Robust convex quadratically constrained programs*, Math. Program., 97 (2003), pp. 495–515. New trends in optimization and computational algorithms (NTOC 2001) (Kyoto).

[14] ———, *Robust portfolio selection problems*, Math. Oper. Res., 28 (2003), pp. 1–38.

[15] L. GRIPPO AND M. SCIANDRONE, *Globally convergent block-coordinate techniques for unconstrained optimization*, Optim. Methods Softw., 10 (1999), pp. 587–637.

[16] L. GRIPPO AND M. SCIANDRONE, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.

[17] C. HELMBERG AND F. RENDL, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696 (electronic).

[18] L. LOVÁSZ, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[19] Z. Q. LUO AND P. TSENG, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[20] S. MA, D. GOLDFARB, AND L. CHEN, *Fixed point and Bregman iterative methods for matrix rank minimization*, tech. rep., Department of IEOR, Columbia University, 2008.

[21] J. MALICK, J. POVH, F. RENDL, AND A. WIEGELE., *Regularization methods for semidefinite programming*, tech. rep., Alpen-Adria-Universität Klagenfurt, Austria, 2008.

[22] M. J. D. POWELL, *On search directions for minimization algorithms*, Math. Programming, 4 (1973), pp. 193–201.

[23] R. T. ROCKAFELLAR, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*,

Math. Oper. Res., 1 (1976), pp. 97–116.

[24] X.-C. Tai and J. Xu, *Global and uniform convergence of subspace correction methods for some convex optimization problems*, Math. Comp., 71 (2002), pp. 105–124 (electronic).

[25] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

[26] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Math. Program., 117 (2009), pp. 387–423.

[27] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[28] H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of semidefinite programming*, International Series in Operations Research & Management Science, 27, Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications.

[29] F. Zhang, ed., *The Schur complement and its applications*, vol. 4 of Numerical Methods and Algorithms, Springer-Verlag, New York, 2005.

[30] X. Zhao, D. Sun, and K. Toh, *A newton-cg augmented lagrangian method for semidefinite programming*, tech. rep., Department of Mathematics, National University of Singapore, 2008.